

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**MĚŘENÍ KVALITATIVNÍCH PARAMETRŮ V PŘÍSTUPOVÉ  
SÍTI**

MEASUREMENT OF QUALITATIVE PARAMETERS IN THE ACCESS NETWORK

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Tomáš Kejík

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. David Grenar

**BRNO 2021**

# Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

**Student:** Tomáš Kejík

**ID:** 211258

**Ročník:** 3

**Akademický rok:** 2020/21

**NÁZEV TÉMATU:**

## Měření kvalitativních parametrů v přístupové síti

### POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je provést teoretický rozbor problematiky přístupové sítě s vysokou kapacitou a jejích klíčových přenosových parametrů. Důraz bude kladen na problematiku zpoždění v přístupových sítích s vysokou kapacitou. Výstupem práce pak bude charakterizace multimediálních přenosů v přístupové síti a vypracování testovacího scénáře v simulačním prostředí. Na základě osvojených znalostí bude navrženo řešení s využitím open source testovacího generátoru síťového provozu a následně bude provedena analýza přenosových parametrů přístupové sítě.

### DOPORUČENÁ LITERATURA:

[1] SPORTACK, Mark A. Směrování v sítích IP: [autorizovaný výukový průvodce: samostudium: kompletní zdroj informací o směrování a protokolech v sítích IP]. Brno: Computer Press, 2004. Cisco systems. ISBN 80 -25-01-7-4.

[2] LAFATA, Pavel a Jiří VODRÁŽKA. Optické přístupové sítě a přípojky FTTx. Praha: České vysoké učení technické v Praze, 2014. ISBN 978-800-1054-635.

**Termín zadání:** 1.2.2021

**Termín odevzdání:** 31.5.2021

**Vedoucí práce:** Ing. David Grenar

**prof. Ing. Jiří Mišurec, CSc.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Tématem bakalářské práce je „Měření kvalitativních parametrů v přístupové síti“. Cílem práce byla charakterizace multimediálních přenosů v přístupové síti a vytvoření funkčního síťového generátoru. Teoretická část je věnována rozboru klíčových přenosových parametrů, které mají vliv na výslednou kvalitu síťového přenosu. Dále jsou rozebrány použité síťové protokoly, které se využívají pro síťový a multimediální přenos. Simulace byla uskutečněna pomocí programu GNS3. Následně byla změřena přenášená data. Jednotlivé parametry určují kvalitu videoobsahu. Poté byly analyzovány nežádoucí jevy ovlivňující funkčnost výsledného přenosu. Dále se praktická část zabývá zhotovením síťového generátoru vytvořeného v jazyce Bash a také vytvořením skriptu za účelem usnadnění zapínání generátoru pro různý počet klientů. Posledním krokem bylo vyhodnocení přenosových parametrů.

## KLÍČOVÁ SLOVA

Bitová rychlost, GNS3, OTT, propustnost, RTP, zpoždění, ztrátovost paketů

## ABSTRACT

The theme of this bachelor's thesis is „Measuring of qualitative parameters in an access network.“ The aim of this paper is a characterization of multimedia broadcasts in access networks and the creation of a functional network generator. The theoretical part of the paper is dedicated to analyzing key broadcasting parameters, which affect the resulting quality of network broadcast. Furthermore, used network protocols that are given for network and multimedia broadcast are analyzed. Simulation is done by using of GNS3 software. After that, transferred data were measured. Side effects affecting the functionality of the resulting transmission were subsequently analyzed. Another part of the paper is the practical part, which is focused on creation of a network generator by using the Bash language and making a script to facilitate turning on the generator for different counts of clients. At the end of the paper, the broadcast parameters were evaluated.

## KEYWORDS

Bit rate, GNS3, OTT, throughput, RTP, delay, packet loss

KEJÍK, Tomáš. *Měření kvalitativních parametrů v přístupové síti*. Brno, 2020, 77 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. David Grenar

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Měření kvalitativních parametrů v přístupové síti“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Davidu Grenarovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále chci poděkovat všem blízkým, kteří v době tvorby této práce stáli při mně a byli mi oporou.

# Obsah

|  |           |
|--|-----------|
| <b>Úvod</b>  | <b>11</b> |
| <b>1 TCP/IP</b>  | <b>12</b> |
| 1.1 Vrstva síťového rozhraní . . . . .                 | 13        |
| 1.2 Internetová vrstva . . . . .                       | 13        |
| 1.3 Transportní vrstva . . . . .                       | 13        |
| 1.3.1 Transmission Control Protocol . . . . .          | 13        |
| 1.3.2 User Datagram Protocol . . . . .                 | 14        |
| 1.4 Aplikační vrstva . . . . .                         | 14        |
| <b>2 Síťové protokoly</b>                              | <b>15</b> |
| 2.1 Internet Protocol . . . . .                        | 15        |
| 2.2 Řídící protokoly . . . . .                         | 15        |
| 2.3 Multimediální služby . . . . .                     | 16        |
| 2.3.1 Real-Time Transport Protocol . . . . .           | 16        |
| 2.3.2 MPEG-Transport stream a komprese dat . . . . .   | 18        |
| <b>3 Přenosové parametry</b>                           | <b>20</b> |
| 3.1 Zpoždění . . . . .                                 | 20        |
| 3.1.1 Zpoždění v přístupové síti . . . . .             | 20        |
| 3.2 Propustnost . . . . .                              | 21        |
| 3.3 Šířka pásma . . . . .                              | 21        |
| 3.4 Přenosová rychlost . . . . .                       | 21        |
| 3.5 Kolísání zpoždění . . . . .                        | 21        |
| 3.6 Ztrátovost paketů . . . . .                        | 22        |
| 3.7 Kvalita služeb . . . . .                           | 22        |
| <b>4 Simulační prostředí</b>                           | <b>23</b> |
| 4.1 Dynamips . . . . .                                 | 23        |
| 4.2 QEMU . . . . .                                     | 24        |
| 4.3 VirtualBox a WMware . . . . .                      | 24        |
| 4.4 DOCKER . . . . .                                   | 24        |
| 4.5 Wireshark . . . . .                                | 25        |
| <b>5 Simulační scénář</b>                              | <b>26</b> |
| 5.1 Postup při vytváření topologie . . . . .           | 26        |
| 5.2 Nástroje pro měření a generování provozu . . . . . | 27        |
| 5.2.1 Ostinato . . . . .                               | 28        |

|          |  |           |
|----------|--|-----------|
| 5.2.2    | Síťové nástroje příkazového řádku . . . . .                | 28        |
| <b>6</b> | <b>Multimediální přenosy</b>                               | <b>30</b> |
| 6.1      | RTP stream na lince 1000BASE-T . . . . .                   | 32        |
| 6.1.1    | RTP stream video 360p . . . . .                            | 33        |
| 6.1.2    | RTP stream video 480p . . . . .                            | 34        |
| 6.1.3    | RTP stream video 720p . . . . .                            | 35        |
| 6.1.4    | RTP stream video 1080p . . . . .                           | 37        |
| 6.1.5    | Souhrnné porovnání . . . . .                               | 38        |
| 6.1.6    | Vývoj zpoždění v čase . . . . .                            | 40        |
| 6.2      | RTP stream na lince 100BASE-TX . . . . .                   | 42        |
| 6.2.1    | RTP stream pro video 360p . . . . .                        | 42        |
| 6.2.2    | RTP stream pro video 480p . . . . .                        | 43        |
| 6.2.3    | RTP stream pro video 720p . . . . .                        | 44        |
| 6.2.4    | RTP stream pro video 1080p . . . . .                       | 46        |
| 6.2.5    | Souhrnné porovnání . . . . .                               | 47        |
| 6.3      | RTP stream na lince 10BASE-T . . . . .                     | 49        |
| 6.3.1    | RTP stream video 360p . . . . .                            | 50        |
| 6.3.2    | RTP stream video 480p . . . . .                            | 52        |
| 6.3.3    | RTP stream video 720p . . . . .                            | 54        |
| 6.3.4    | RTP stream video 1080p . . . . .                           | 55        |
| 6.4      | Porovnání kodeků a jejich přenosových parametrů . . . . .  | 56        |
| 6.4.1    | H.264 . . . . .  | 56        |
| 6.4.2    | MPEG-2 . . . . .   | 57        |
| 6.4.3    | MPEG-4 . . . . .   | 59        |
| 6.4.4    | Souhrnné porovnání . . . . .                               | 60        |
| 6.4.5    | Změření bitové rychlosti pro různý počet klientů . . . . . | 63        |
| <b>7</b> | <b>Generátor síťového provozu</b>                          | <b>65</b> |
| 7.1      | Simulace síťového provozu . . . . .                        | 68        |
| 7.1.1    | UDP simulace . . . . .                                     | 69        |
| 7.1.2    | RTP simulace . . . . .                                     | 70        |
|          | <b>Závěr</b>   | <b>72</b> |
|          | <b>Literatura</b>  | <b>74</b> |
|          | <b>Seznam symbolů, veličin a zkratk</b>                    | <b>76</b> |



# Seznam obrázků

|      |  |    |
|------|--|----|
| 1.1  | TCP/IP . . . . .   | 12 |
| 2.1  | RTP hlavička . . . . .   | 17 |
| 5.1  | Topologie sítě . . . . .   | 26 |
| 5.2  | Spojení přes NAT . . . . .   | 27 |
| 6.1  | Linka 1000BASE-T a 100BASE-TX . . . . .  | 31 |
| 6.2  | Bitová rychlost v čase (360p) . . . . .  | 33 |
| 6.3  | Zpoždění a jeho kolísání v závislosti na čase (360p) . . . . .                             | 34 |
| 6.4  | Bitová rychlost v čase (480p) . . . . .  | 35 |
| 6.5  | Zpoždění a jeho kolísání v závislosti na čase (480p) . . . . .                             | 35 |
| 6.6  | Bitová rychlost v čase (720p) . . . . .  | 36 |
| 6.7  | Zpoždění a jeho kolísání v závislosti na čase (720p) . . . . .                             | 36 |
| 6.8  | Bitová rychlost v čase (1080p) . . . . .   | 37 |
| 6.9  | Zpoždění a jeho kolísání v závislosti na čase (1080p) . . . . .                            | 38 |
| 6.10 | Souhrnný graf pro jednotlivá videa . . . . .   | 40 |
| 6.11 | Souhrnný graf pro kolísání zpoždění v závislosti na čase pro jednotlivá<br>videa . . . . . | 40 |
| 6.12 | Bitová rychlost v čase (360p) - Fast Ethernet . . . . .                                    | 42 |
| 6.13 | Zpoždění a jeho kolísání v závislosti na čase (360p) - Fast Ethernet . . . . .             | 43 |
| 6.14 | Bitová rychlost v čase (480p) - Fast Ethernet . . . . .                                    | 44 |
| 6.15 | Zpoždění a jeho kolísání v závislosti na čase (480p) - Fast Ethernet . . . . .             | 44 |
| 6.16 | Bitová rychlost v čase (720p) - Fast Ethernet . . . . .                                    | 45 |
| 6.17 | Zpoždění a jeho kolísání v závislosti na čase (720p) - Fast Ethernet . . . . .             | 45 |
| 6.18 | Bitová rychlost v čase (1080p) - Fast Ethernet . . . . .                                   | 46 |
| 6.19 | Zpoždění a jeho kolísání v závislosti na čase (1080p) - Fast Ethernet . . . . .            | 47 |
| 6.20 | Souhrnný graf pro jednotlivá videa - Fast Ethernet . . . . .                               | 49 |
| 6.21 | Souhrnný graf pro kolísání zpoždění v závislosti na čase - Fast Ethernet . . . . .         | 49 |
| 6.22 | Počet přenesených paketů v čase pro video rozlišení 360p . . . . .                         | 50 |
| 6.23 | Kolísání zpoždění pro video rozlišení 360p . . . . .                                       | 50 |
| 6.24 | Závislost zpoždění a jeho kolísání na čase pro 2 klienty (video 360p) . . . . .            | 51 |
| 6.25 | Počet přenesených paketů v čase pro video rozlišení 480p . . . . .                         | 52 |
| 6.26 | Kolísání zpoždění pro video rozlišení 480p . . . . .                                       | 52 |
| 6.27 | Závislost zpoždění a jeho kolísání na čase pro 2 klienty (video 480p) . . . . .            | 53 |
| 6.28 | Počet přenesených paketů v čase pro video rozlišení 720p . . . . .                         | 54 |
| 6.29 | Kolísání zpoždění pro video rozlišení 720p . . . . .                                       | 54 |
| 6.30 | Počet přenesených paketů v čase pro video rozlišení 1080p . . . . .                        | 55 |
| 6.31 | Kolísání zpoždění pro video rozlišení 1080p . . . . .                                      | 55 |
| 6.32 | Bitová rychlost v čase (kodek H.264) . . . . .   | 57 |

|      |   |    |
|------|---|----|
| 6.33 | Zpoždění a jeho kolísání v závislosti na čase (kodek H.264) . . . . .                       | 57 |
| 6.34 | Bitová rychlost v čase (kodek MPEG-2) . . . . .   | 58 |
| 6.35 | Zpoždění a jeho kolísání v závislosti na čase (kodek MPEG-2) . . . . .                      | 59 |
| 6.36 | Bitová rychlost v čase (kodek MPEG-4) . . . . .   | 60 |
| 6.37 | Zpoždění a jeho kolísání v závislosti na čase (kodek MPEG-4) . . . . .                      | 60 |
| 6.38 | Souhrnný graf pro bitovou rychlost a zpoždění v čase pro jednotlivé<br>kodeky . . . . .     | 62 |
| 6.39 | Souhrnný graf pro kolísání zpoždění v závislosti na čase pro jednotlivé<br>kodeky . . . . . | 63 |
| 6.40 | Bitová rychlost v čase pro různý počet klientů (kodek H.264) . . . . .                      | 64 |
| 6.41 | Bitová rychlost v čase pro různý počet klientů (kodek MPEG-4) . . . . .                     | 64 |
| 7.1  | Velikost paketů pro UDP simulaci . . . . .  | 69 |
| 7.2  | Propustnost linky pro UDP simulaci - upload . . . . .                                       | 70 |
| 7.3  | Propustnost linky pro UDP simulaci - download . . . . .                                     | 70 |
| 7.4  | Velikost paketů pro RTP simulaci . . . . .  | 71 |
| 7.5  | Propustnost linky pro RTP simulaci - upload . . . . .                                       | 71 |
| 7.6  | Propustnost linky pro RTP simulaci - download . . . . .                                     | 71 |

# Seznam tabulek

|      |   |    |
|------|---|----|
| 6.1  | Parametry pro video 360p . . . . .                              | 31 |
| 6.2  | Parametry pro video 480p . . . . .                              | 31 |
| 6.3  | Parametry pro video 720p . . . . .                              | 32 |
| 6.4  | Parametry pro video 1080p . . . . .                             | 32 |
| 6.5  | Změřené parametry pro přenos videa 360p . . . . .               | 38 |
| 6.6  | Změřené parametry pro přenos videa 480p . . . . .               | 39 |
| 6.7  | Změřené parametry pro přenos videa 720p . . . . .               | 39 |
| 6.8  | Změřené parametry pro přenos videa 1080p . . . . .              | 39 |
| 6.9  | Výpočet logaritmické spojnice trendu pro 1080p a 720p . . . . . | 41 |
| 6.10 | Výpočet logaritmické spojnice trendu pro 480p a 360p . . . . .  | 41 |
| 6.11 | Změřené parametry video 360p - Fast Ethernet . . . . .          | 47 |
| 6.12 | Změřené parametry video 480p - Fast Ethernet . . . . .          | 48 |
| 6.13 | Změřené parametry video 720p - Fast Ethernet . . . . .          | 48 |
| 6.14 | Změřené parametry video 1080p - Fast Ethernet . . . . .         | 48 |
| 6.15 | Změřené parametry pro kodek H.264 . . . . .                     | 61 |
| 6.16 | Změřené parametry pro kodek MPEG-2 . . . . .                    | 61 |
| 6.17 | Změřené parametry pro kodek MPEG-4 . . . . .                    | 61 |

# Úvod

Po celém světě poskytovatelé internetu čelí výzvám zlepšovat kvalitu služeb při zvyšování kapacity sítě a při zavádění nových služeb. Vznikají problémy s přetížením sítě a následně je vlivem nedostatečných přenosových parametrů degradována QoS. Při poskytování multimediálních služeb je pro koncového uživatele nepodstatné, jaké technologie jsou použity, aby mohl být přenos realizován. Koncový uživatel pouze zhodnotí, zda kvalita služby odpovídá jeho představám, či nikoliv. Správci sítě tak musí neustále monitorovat různé části sítě a případné problémy odstraňovat, aby byly splněny požadavky vůči zákazníkům.

Tato práce se zabývá měřením kvalitativních parametrů, které mají vliv na přenos. Základem počítačových sítí je TCP/IP model, který obsahuje sadu protokolů pro komunikaci. Tento model je popsán v první teoretické části práce, obsahuje základní informace o problematice fungování sítí. Architektura TCP/IP je rozdělena na jednotlivé vrstvy, které jsou následně rozebrány. Druhá část této práce je věnována jednotlivým protokolům, které se využívají ke správnému fungování sítě. Tento teoretický rozbor se týká řídicího protokolu, dále protokolu využívajícího se v komunikaci a také protokolu sloužícího k uskutečnění multimediálních přenosů. Nej důležitější částí jsou přenosové parametry, které ovlivňují přenos dat, kvalitu služeb a kvalitu zážitků. V poslední teoretické části jsou popsány programy, které byly využity pro praktickou část. První zmíněný program je GNS3. V tomto programu je vytvořena topologie pro simulaci přenosu. Dále jsou tu emulátory, které jsou využívány v GNS3, Dynamips, QEMU, VirtualBox, VMware a Docker. Jako poslední je popsán program Wireshark, který je používán pro analýzu a zachytávání přenosu.

Praktická část obsahuje informace o tvorbě topologie. Dále jsou zde zmiňovány nástroje pro generování a měření provozu. Prvním cílem praktické části byla charakterizace multimediálních přenosů. K samotnému přenosu byla vybrána čtyři videa různých kvalit. Multimediální přenos byl uskutečněn na třech různých linkách, které mají různou propustnost dat. Jednotlivé výsledky jsou analyzovány a porovnávány. Multimediální přenosy jsou také charakterizovány z hlediska použitých kodeků a výsledné parametry přenosu jsou také analyzovány.

Druhý cíl v praktické části sledoval zhotovení síťového generátoru provozu. Tento generátor byl vytvořen pomocí Bash skriptu. K usnadnění zapínání simulace byl také zhotoven druhý skript, který umožňuje spustit generování pro různý počet klientů. Simulace byla uskutečněna pomocí UDP a RTP protokolu. Následně došlo k vyhodnocení výsledků simulace.

V závěru jsou zhodnoceny výsledky.

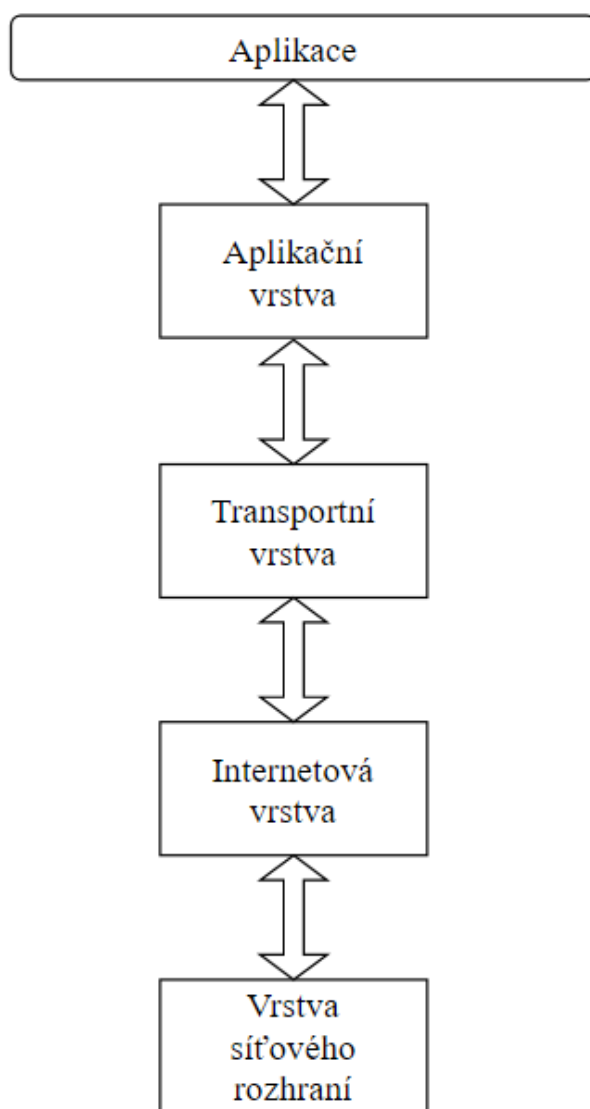
# 1 TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) je síťová architektura, která slouží pro komunikaci v počítačové síti. Protokoly, které má ve svém názvu, tvoří její jádro. Rodina protokolů TCP/IP se nezabývá až na výjimky fyzickou a linkovou vrstvou. TCP/IP je hlavním protokolem pro komunikaci v celém Internetu.

Tento model se skládá ze 4 vrstev:

1. Vrstva síťového rozhraní (Network Interface Layer)
2. Internetová vrstva (Internet Layer)
3. Transportní vrstva (Transport Layer)
4. Aplikační vrstva (Application Layer) [1, 2, 3]

Obr. 1.1: TCP/IP



## 1.1 Vrstva síťového rozhraní

Často je také nazývána jako Ethernetová vrstva. Svými funkcemi odpovídá dvěma nejnižším vrstvám fyzické a spojové. Vrstva síťového rozhraní je nejnižší vrstvou a je specifikována až podle přenosového média. Na této vrstvě probíhá komunikace mezi sousedními počítači v lokální síti. Je skutečným rozhraním síťového hardware. Specifikuje způsob využití sítě pro vysílání datových jednotek. Probíhá zde adresace, formát datových jednotek a následné formátování do rámců. [2, 3]

## 1.2 Internetová vrstva

Jedná se o druhou vrstvu síťového modelu, která již není závislá na přenosové technologii. Označuje se jako IP vrstva, protože je realizována pomocí protokolu IP (IPv4 nebo IPv6). Omezuje se na tuto službu bez spojení. Tato vrstva má za úkol především směrování paketů. Základní jednotkou přenosu je síťový paket, který se balí do datového rámce. Síťová vrstva zabezpečuje přenos dat mezi vzdálenými počítači WAN. Jednotlivé dílčí sítě mohou být odlišné a síťová vrstva se s tím musí vypořádat. Zde se také nachází režijní protokol ICMP, který bude popsán níže. [1, 2, 3]

## 1.3 Transportní vrstva

Transportní vrstva je třetí vrstvou TCP/IP modelu. Někdy je také označována jako TCP vrstva. Hlavní úkol této vrstvy spočívá v zajištění přenosu mezi koncovými účastníky end-to-end komunikace. Tuto službu poskytuje počítačovým aplikacím. Transportní vrstva se zcela spoléhá na služby nižších vrstev. Zajištění spolehlivosti je řešeno touto vrstvou a nejpoužívanější protokol je TCP. Dále je tu také protokol UDP, který naopak nabízí nespolehlivou službu a bude využíván v praktické části na streamování multimediálního obsahu po síti. Protokoly TCP a UDP zajišťují spojení mezi aplikacemi vzdálených počítačů. Pro adresaci jsou využívány porty. Port je 16bitové kladné číslo a je přiděleno každé komunikující straně. [1, 3]

### 1.3.1 Transmission Control Protocol

Transmission Control Protocol (TCP) zajišťuje spolehlivou a spojovanou službu. Základní přenosová jednotka se nazývá segment. Příjemce používá potvrzovací mechanismy, které slouží k ověření, zda skutečně došlo k úspěšnému odeslání dat. Spojovaná služba znamená, že se mezi dvěma aplikacemi vytvoří spojení (virtuální okruh). Virtuální okruh je plně duplexní. Přenášené bajty jsou číslovány. Garantuje tedy doručení segmentů ve správném pořadí. Poškozená, nebo ztracená data

příjemce vyžaduje znovu. Tento protokol zajišťuje vyšší režii (řízení propustnosti a řízení toku dat), avšak v porovnání s protokolem UDP je pomalejší. TCP protokol je pro realtime přenos nevhodný. Opakování přenosu v reálném čase není možné. Využívá se v aplikacích, které nejsou tak časově kritické. [1, 3, 4]

### **1.3.2 User Datagram Protocol**

User Datagram Protocol (UDP) zajišťuje nespolehlivou a nespojovanou službu. Přenášené jednotky nazýváme datagramy. Nezaručuje správné pořadí doručených datagramů, ani nezaručuje, že se daný datagram neztratí a nepošle víckrát. Oproti TCP je protokol UDP velice jednoduchý. Málo funkcí značí menší režii přenosu a tím menší zpoždění. UDP umožňuje vysílání zpráv na multicastovou adresu. Využívá se u multimediálních přenosů, u nichž je důležitější rychlost odesílání dat před režii provozu. Multimediální přenosy jsou na spolehlivost nenáročné, ale zároveň časově citlivé. UDP protokol nedefinuje žádné metody, jak přenášet multimediální data ani nezaručuje přenos všech odeslaných dat. [3, 4]

## **1.4 Aplikační vrstva**

Jedná se o nejvyšší vrstvu jejímiž entitami jsou přímo aplikace. Komunikace aplikační vrstvy probíhá přímo s transportní vrstvou. Na aplikační vrstvě probíhá komprese dat, signalizace a přenos dat. Prezentační a relační služby si aplikace musí realizovat sama a pokud nejsou potřeba, je ušetřena režie. Aplikační vrstva poskytuje aplikacím různé služby, například přenos zpráv, identifikace komunikujících partnerů. Na této vrstvě najdeme mnoho síťových protokolů. Mezi nejznámější patří protokol Hypertext Transfer Protocol (HTTP), základní přenosový protokol webového prostředí a nebo také Dynamic Host Control Protocol (DHCP), protokol pro dynamickou správu IP adres. Také na této vrstvě lze nalézt protokol Real Time Protocol (RTP) používaný pro přenos dat v reálném čase. [3]

## 2 Síťové protokoly

Protokoly definují, jakým způsobem si směrovače vyměňují informace, a umožňují směrovačům dynamicky zjišťovat informace o ostatních sítích, do nichž lze směrovat pakety.

### 2.1 Internet Protocol

Internet protocol (IP) je hlavní protokol síťové vrstvy. Slouží ke komunikaci v počítačových sítích a v celosvětové síti Internetu. Zajišťuje přenos IP paketů a směrování po síti. Funguje jako nespolehlivý (negarantuje doručení a nepoškozenost dat, nepoužívá potvrzení dat, nepodporuje řízení toku) a nespojovaný (negarantuje pořadí a dobu doručení, nečísluje přenášené pakety) protokol. Velikost IP paketu je proměnná, maximální velikost může být 65535 bajtů a minimální velikost je 576 bajtů bez toho, aniž by docházelo k fragmentaci. Podporuje řídicí protokoly jako ICMP a IGMP. Existují dvě verze IP protokolu, verze číslo 4 a 6.

IPv4 má 32bitovou adresu. Používá tedy pro adresaci 32 bitů a velikost adresního prostoru je 4,3 miliard adres. Adresu můžeme zapsat binárně či dekadicky. Masku adresy se používá k určení sítě z hostitelské části.

Důvodem zavádění nového protokolu byl zmenšující se adresní prostor. Došlo ke zvětšení adresního prostoru ( $3,4 \cdot 10^9$  adres). Mezi některé novinky, které přinesl tento protokol, patří: nové podpůrné protokoly, optimalizace pro vysokorychlostní směrování a podpora služby QoS. IPv6 má 128bitovou adresu, která je zapsána hexadecimálně (0-9 a A-F). IPv6 a IPv4 nejsou mezi sebou kompatibilní. [5, 6]

### 2.2 Řídící protokoly

Internet Control Message Protocol (ICMP) je režijní protokol Internetu, který nepřenáší žádná uživatelská data. Jelikož IP protokol neobsahuje žádné hlášení a opravu chyb, neumožňuje ani testování dostupnosti, tak právě tento protokol řeší tyto uvedené nedostatky. Obsah protokolu ICMP se přímo přenáší v IP datagramu, který chybovou zprávu vyvolal.

Máme dvě skupiny ICMP zpráv. První skupina pojednává o hlášení chyb:

- Typ 3 Nedoručitelný datagram - při směrování dojde k zahození paketu, například z důvodu, že se neví, kam dále v souvislosti s fragmentací paket směřuje.
- Typ 4 Snížení rychlosti odeslání - řízení toku z důvodu zahlcení sítě.
- Typ 5 Přesměrování - používá se při směrování ven ze sítě (např. změna výchozí brány)



- Typ 11 Vypršení doby života - TTL je 8bitové číslo, které se každým skokem snižuje. Pokud dojde k situaci  $TTL = 0$  je paket zahozen.
- Typ 12 Problém s parametry - v případě, že záhlaví paketu obsahuje nepřesně definovanou informaci.

Druhá skupina jsou informační zprávy:

- Typ 8 a Typ 0 Žádost o odezvu a odpověď - slouží k ověření dostupnosti zařízení. Nejčastěji toho využívá aplikace ping.
- Typ 13 a Typ 14 Požadavek na časové měřítko a odpověď - využívá se u měření zpoždění na přenosové trase v režimu RTT.

ICMP je využíván programem Ping, který slouží k určení, zda je konkrétní IP adresa dostupná, nebo zda je dostupný daný hostitel, také slouží k měření zpoždění. Program ping spustíme pomocí příkazového řádku. Můžeme zobrazit počet odeslaných paketů spolu s počtem přijatých, popř. počet ztracených paketů. [3]

## 2.3 Multimediální služby

Při streamování, dekódování videa a zvuku přes IP síť je nutné zajistit kvalitu služeb (Qos) v reálném čase. Pro přenos multimediálních služeb je potřeba velká šířka pásma. V sítích vzniká zpoždění a ztráta paketů. Tyto nežádoucí jevy negativně ovlivňují kvalitu přenosu. Pro streamování se využívají různé protokoly jako například RTP.

### 2.3.1 Real-Time Transport Protocol

Real-Time Transport Protocol (RTP) je definovaný v RFC 3550. Tento protokol poskytuje služby typu end-to-end pro přenos dat v reálném čase, jako přenos videa a zvuku. Mezi tyto služby patří identifikace typu zátěže, časové razítko, sekvenční číslování a monitorování provozu. Aplikace využívají RTP nad UDP, z něhož se využívá multiplexování a kontrolní součet. RTP sám sobě neposkytuje mechanismus, díky němuž by bylo zajištěno správné či včasné doručení dat nebo další poskytování kvality služeb. To je řešeno na nižších vrstvách. Sekvenční čísla obsažená v RTP umožňují přijímací straně zkontrolovat pořadí paketů. Sekvenční čísla také slouží k určení správného umístění paketu při dekódování, aniž by bylo nutné provádět dekódování postupně. [7]

#### Hlavička RTP

V této podkapitole je charakterizována hlavička RTP protokolu.

**V - Version (2b)** Identifikuje verzi protokolu RTP.

**P - Padding (1b)** Informace o doplňkových oktetech na konci užitečných dat. Nejsou součástí přenášené informace. Při dekódování paketu se tyto oktety ignorují.

**X - Extension (1b)** Hlavička RTP musí být rozšířena, pokud je bit nastaven na hodnotu 1.

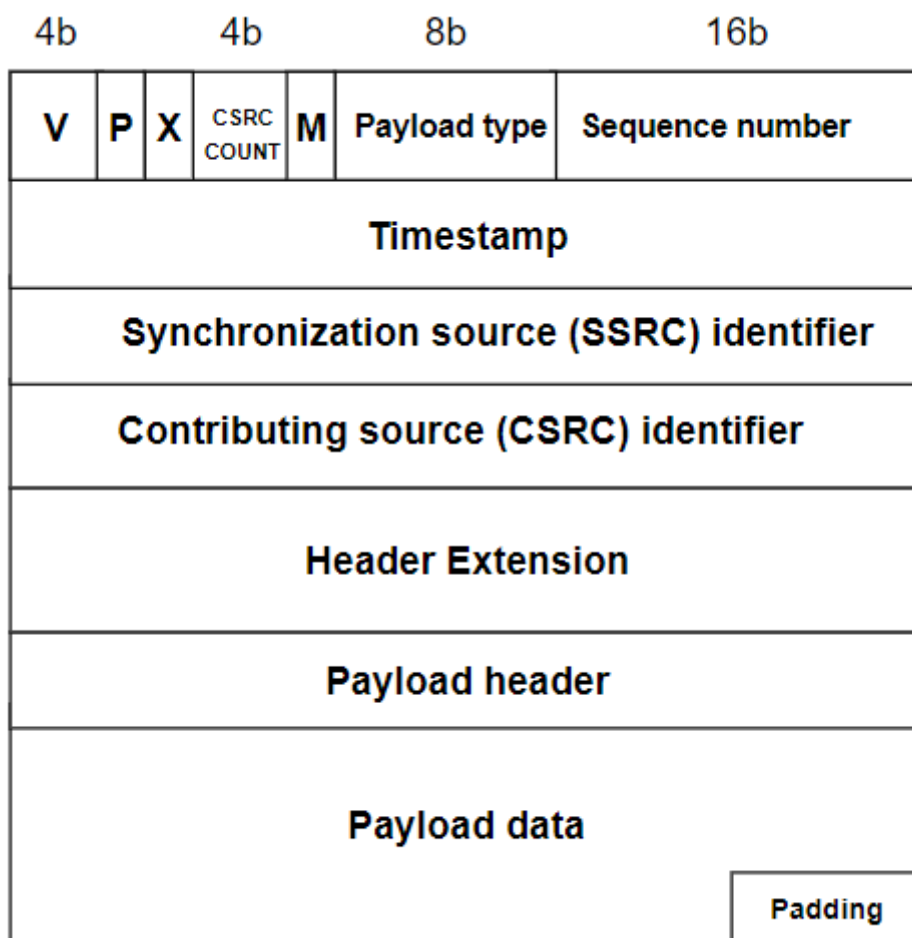
**CSRC Count (4b)** Udává počet CSRC identifikátorů, které rozšiřují hlavičku.

**M - Marker (1b)** Používá se např. jako oznámení nějaké události (poslední přenášený paket).

**Payload type (7b)** Identifikace typu zátěže přenášené v RTP paketu. Typy zátěže jsou rozlišovány podle datových typů.

**Sequence number (16b)** Poskytuje přehled o doručených paketech. Díky sekvencovnímu číslu lze snadno dohledat ztracené pakety.

Obr. 2.1: RTP hlavička



[4]

**Timestamp (32b)** Používá se především při plánování multimediálního streamu. Časové razítko identifikuje okamžik prvního oktetu v RTP paketu.

**Synchronization Source (SSRC) identifier (32b)** Slouží k identifikaci zdroje multimediálního streamu. Pro každého účastníka je vygenerováno náhodné číslo.

**Contributing Source (CSRC) identifier (32b)** Je využíván v případě, jestliže jsou data RTP paketu generována více zdroji. Každý ze zdrojů obdrží CSRC identifikátor.

**Header Extension** Rozšíření je povoleno, pokud hodnota u X je nastavena na 1. Header Extension může být libovolně velká, počínaje od 16b.

**Payload header** Informuje o každé zátěži.

**Payload data** Poslední část RTP paketu. Velikost a formát zátěže závisí na nastavených parametrech. [4]

## Zpoždění a jeho kolísání

V případě RTP protokolu se pro jitter (kolísání zpoždění) jedná o statistický odhad odchylky v časovém intervalu mezi doručováním RTP paketů, měřeného a zapsaného v hlavičce (položka *Timestamp*).

Intervalový jitter  $J$  je definován jako střední odchylka (vyhlazená absolutní hodnota) rozdílu  $D$  v rozestupu paketů u přijímače ve srovnání s odesílatelem pro pár zvolených paketů. V rovnici uvedené níže pro pár paketů je relativní doba přechodu rozdílem mezi časovým měřítkem RTP paketů a hodinami přijímače v době přijetí. Měření probíhá ve stejných jednotkách.

Pokud je  $S_i$  časové razítko RTP z paketu  $i$  a  $R_i$  značí čas příchodu do jednotek RTP časového razítka pro paket  $i$ . Poté platí pro dva pakety  $i, j$  zpoždění  $D$ , které lze vyjádřit rovnicí 2.1.

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (2.1)$$

Kolísání zpoždění by mělo být počítáno nepřetržitě (datový paket  $i$  je přijat ze zdroje *SSRC*). Použije se rozdíl zpoždění  $D$  pro tento paket a také pro následující  $i-1$  dle pořadí příchodu (není nutné zachovat pořadí). Pro tento výpočet platí vztah 2.2.

$$J(i) = J(i-1) + (|D(i-1, i)| - J(i-1))/16 \quad (2.2)$$

Při každém vydání zprávy o příjmu je hodnota  $J$  vzorkována. Tento algoritmus je optimálním odhadem prvního řádu a parametr zisku  $1/16$  poskytuje dobrý poměr redukce šumu při zachování přiměřené rychlosti konvergence.[7]

## 2.3.2 MPEG-Transport stream a komprese dat

Velmi populárním standardem se stalo digitální video. Má velice širokou škálu využití, pro přenos videa pro mobilní komunikaci, televizi nebo pro datová média (CD,

DVD). Nevýhodou digitálního videa je jeho velikost. Nekomprimovaná videa o době trvání 1 sekundy mají velikost přibližně 32 MB, proto je potřeba video komprimovat pomocí vhodného kompresního standardu. Používají se standardy, které jsou vyvíjeny organizacemi ITU-T (H.261, H.263 a H.264) a ISO/IEC (MPEG-1, MPEG-2 a MPEG-4).

Standardy MPEG jsou aktuálně jedny z nejpoužívanějších standardů a jsou založeny na stejném principu. Používají se pro kompresi videa a audia. Takováto komprese je ztrátová tzn., že dekódovaná data nejsou stejné jako originál. Může nastat, jak časová redundance, tak i prostorová. Pro kompresi časové redundance je využívána podobnost sousedních snímků. Využívá se tzv. predikce snímku. Současný snímek může být predikcí současného nebo následujícího snímku. Poté je kódován pouze rozdíl mezi predikovaným snímkem a tím skutečným.

Standard MPEG-2 byl určen pro ztrátové prostředí. Obsahuje mechanismy pro zabezpečení proti chybovosti při vysílání. Tento standard definuje dva typy streamů. Pro přenos sítí se používá transportní stream. Transportní stream je určen pro ztrátové prostředí, což odpovídá síťovému přenosu, kde dochází k různým ztrátám. Lze ho využít i pro přenos více videí v jednom streamu. Pakety mají velikost 188 B a jsou označeny PID (Program identifier), hlavička má velikost 4 B. Základem je tzv. elementární stream (ES), který dále může být rozdělen na paketové elementární streamy (PES). Každý takový PES obsahuje hlavičku (ta obsahuje také časové razítko), podle které se určí začátek paketu. Transportní stream obsahuje nejen multimediální data, ale i metadata. Metadata zahrnují tabulku PAT (Program Association Table). Každá položka PAT odkazuje na konkrétní tabulku PMT (Program Map Table).

MPEG-4 je založen na standardu MPEG-2, ale má efektivnější kompresi. H.264 je standard, který podporuje jak progresivní, tak prokládané video. Kódovaný obraz je složen z makrobloků, které tvoří části snímků (slice). Je umožněna intrapedikce, jak pro celé makrobloky, tak pro bloky různých velikostí. [4]

## 3 Přenosové parametry

Mezi základní přenosové parametry patří zpoždění, kolísání zpoždění, propustnost, přenosová rychlost a ztrátovost paketů. Prostřednictvím těchto parametrů je určena výkonnost a efektivita sítě.

### 3.1 Zpoždění

Delay (zpoždění) je doba, po níž trvá přenos paketu od zdroje k cíli. Je závislá na délce trasy, šířce pásma, velikosti odesílaných dat a také na momentálním zatížení sítě. Zpoždění se měří v ms. Do zpoždění se započítávají všechny směrovače, které jsou v rámci přenosové cesty. Všechna tato jednotlivá zpoždění se následně sečtou a vyjde výsledná latence.

Zpoždění dělíme na jednosměrné a obousměrné:

- Jedsměrné je takové zpoždění, které odpovídá definici na začátku kapitoly, tzn. že se jedná o dobu, po kterou trvá přenos paketu od zdroje k cíli.
- Obousměrné zpoždění Round Trip Time (RTT) je definováno jako doba trvání přenosu paketu od zdroje k cíli a následné odeslání zpět. [3]

Hodnoty zpoždění se mohou dynamicky měnit při zpoždění ve frontě na odbavení nebo zpoždění při přepínání v síti. Naopak pevné hodnoty zpoždění nastávají při přenosu nebo při zpoždění kódováním a sestavováním paketů (tzv. paketizace). [4]

#### 3.1.1 Zpoždění v přístupové síti

Zpoždění v přístupové síti je způsobeno řadou faktorů:

- Rychlost na poslední míli (nejblíže k uživateli) v přístupové síti je menší než v hlavní síti. To je způsobeno serializačním zpožděním. Serializační zpoždění je čas potřebný k tomu, aby byly přeneseny všechny bity paketu. Pokud jsou posílány různě velké pakety, bude serializační zpoždění přispívat ke kolísání zpoždění. Pokud jsou pakety krátké, zpoždění v obou směrech je velmi malé. Doporučené zpoždění způsobené mediální bránou má být maximálně 30 ms. Pakety mají obvyklou velikost 1500 B, serializace může způsobit velké zpoždění na pomalé síti. [2]
- Zpoždění závisí na momentálním zatížení sítě.
- Některé technologie zahrnují opakování přenosu, což vede ke kolísání zpoždění.
- Pokud technologie používají opravu chyb, tak nastává zpoždění. [8]

## 3.2 Propustnost

Throughput (propustnost) určuje maximální počet paketů, které lze sítí přenést za jednotku času. Tento parametr je reálnou hodnotou a bude mít nižší hodnotu než teoretická šířka pásma. Hodnotu propustnosti může omezit např. formát zprávy, charakter komunikace, nebo vytížení zařízení po přenosové cestě v síti. Závisí na QoS nakonfigurovaných na portech zařízení. Jednotkou propustnosti je b/s. [3, 9]

## 3.3 Šířka pásma

Bandwidth (šířka pásma) označuje celkovou přenosovou kapacitu. Je důležitým faktorem při určování QoS. Jednotkou je bp/s. Šířka pásma je teoretická hodnota, které není možné v koncové aplikaci dosáhnout. Často je zaměňována s dosažitelnou maximální rychlostí. Zvýšení kapacity sítě povede k nižší latenci. Obecně platí, že čím je větší šířka pásma, tím je větší přenosová rychlost, avšak ovlivňuje ji několik faktorů:

- Zpoždění
- Ztráta paketů
- Výkon hardwaru
- Momentální provoz v síti
- Bitové chyby na rozhraní [3, 9]

## 3.4 Přenosová rychlost

Bit rate (přenosová rychlost) udává, jaké množství informace se přenese za jednotku času. Základní přenosovou jednotkou je bit/s. Přenosová rychlost je odvozena podle použitých prvků v síti, může být také dána fyzikálními vlastnostmi použitých přenosových médií. Každý prvek může mít odlišnou přenosovou rychlost, která je daná výrobcem. [3]

## 3.5 Kolísání zpoždění

Jitter (kolísání zpoždění) udává změnu zpoždění mezi příchozími pakety, které následují po sobě. Vzniká v souvislosti rozdílných mechanismů front a v souvislosti se zahlcením sítě. Tento parametr je důležitý především u hlasových multimediálních služeb, kde je potřeba vysílat a přijímat paket každých 20 ms. Při velkém kolísání zpoždění se obraz seká a u hovorů začne vypadávat zvuk. Řešením pro aplikace je zabudování vyrovnávací paměti, do které se ukládá datový tok. [4]

## 3.6 Ztrátovost paketů

Packet Loss (ztrátovost paketů) může nastat z důvodu zahlcení směrovačů, které nemůžou obsloužit v jeden okamžik velký počet příchozích paketů. Směrovače začnou pakety zahazovat. Pokud používáme protokol TCP, tak zahozené pakety jsou odeslány znovu. V takovém případě začne narůstat zpoždění, a to je nevyhovující pro aplikace, které pracují v reálném čase. Pro přenos paketů se používá protokol UDP, jemuž chybí mechanismus pro opětovné odeslání zahozeného paketu. Aplikace v reálném čase tolerují určitou ztrátu paketů. Pozdější doručení paketů by jim nebylo nikterak platné. [2, 4]

## 3.7 Kvalita služeb

Quality of Service (QoS) definována v ITU-T E.800, je schopnost sítě k nastavení dostatečné kvalitní výměny dat u aplikací pracujících v reálném čase. Spočívá především ve zvýhodnění vybraného typu provozu před ostatními a poskytnutí vyšší priority přenosu. Kvalita služby je ovlivněna všemi komponenty (př. servery, přepínači, směrovači), které se nachází v síti. QoS dokáže rozlišit mezi jednotlivými přenosy a každému z nich nastavit jinou kvalitu. Kvalita služby závisí na parametrech jako propustnost, šířka pásma, přenosová rychlost, zpoždění, kolísání zpoždění a na ztrátovosti paketů. Všechny tyto parametry byly popsány výše.

Zpoždění, kolísání zpoždění a šířka pásma významně ovlivňují Quality of Experience (QoE). Doporučení ITU-T Y.1541 definuje výkonnostní cíle pro síť, které doplňují uživatelské požadavky na výkon dle ITU-T G.1010. Y.1541 obsahuje celkem 5 tříd pro QoS. Pro praktickou část je důležitá QoS třída 4. Tato třída určuje doporučení pro streamování videa v reálném čase. Horní mez zpoždění, která je tolerována, je 1 s. Kolísání zpoždění je zde nespecifikováno. [8, 10, 11]

Mechanismy zajištění QoS jsou důležité především u aplikací:

- citlivé na odchylky ve zpoždění (video)
- citlivé na zpoždění (zvuk)
- citlivé na ztrátu paketů (video) [2]

## 4 Simulační prostředí

Graphical Network Simulator 3 (GNS3) je využívána po celém světě k emulaci, konfiguraci, testování a odstraňování problémů se sítěmi. Před zavedením virtualizace se počítačové sítě musely stavět pomocí fyzického hardware. Díky simulátorům lze síť bez problému navrhnout, otestovat ji a tím předejít případným chybám. Umožňuje vytvářet libovolně velké topologie. Autorem simulátoru je Jeremy Grossman, který spolupracoval s lidmi jako Christopher Fillot a Julien Duponchelle. Fillot je tvůrcem emulačního programu Mips (Dynamips) a Duponchelle pomáhal s psaním zdrojového kódu GNS3 a pomohl tak vylepšit software. GNS3 je open source program a jeho používání je zcela zdarma. Jeho zdrojový kód je k dispozici na githubu, každý tak může přispět ke zlepšení programu. I nadále je aktivně rozvíjen celou komunitou lidí, kteří si navzájem pomáhají řešit problémy, předávají si zkušenosti a tak s ním efektivněji pracují. Tento grafický software nabízí virtualizaci reálných hardwarových zařízení (Dynamips), můžeme ho nainstalovat na Linux, Windows a OS X.

GNS3 se skládá ze dvou softwarových komponentů:

- Grafického uživatelského rozhraní (GUI)
- Virtuálního stroje (VM)

V GUI lze vytvářet libovolné topologie, nakonfigurovat síťové parametry a odsimulovat si chování topologie. Musí být vytvořen serverový proces, na kterém simulace bude běžet. Existují tři možnosti serveru:

- Místní server GNS3
- Místní GNS3 VM
- Vzdálený GNS3 VM

Místní server běží lokálně na stejném počítači, na němž je nainstalována GNS3. Druhou možností je použití virtuální stanice, která se dá spustit na některém z virtualizačních programů, jako např. VirtualBox a VMware. Poslední možností je vzdálený přístup. Virtuální stanice je umístěna na jiném místě, než na kterém uživatel pracuje. To je umožněno např. pomocí VMware ESXI. [12, 13, 14]

### 4.1 Dynamips

Od samého začátku je technologie Dynamips součástí GNS3. Je to hypervisor program, který dokáže emulovat směrovače Cisco řady 1700, 2600, 3600, 3700 a 7200 a umožňuje základní přepínání pomocí modulu Etherswitch. Pomocí něj můžeme zkopírovat podporovaný obraz IOS z fyzického síťového zařízení a následně ho použít v GNS3. Dynamips nám umožňuje snadno a rychle konfigurovat modely sítí



s různými druhy rozhraní (SLOT a WAN). Lze přidávat ethernetová rozhraní, přepínací moduly a sériové porty, nebo přidávat či odebírat paměť jednotlivých zařízení v závislosti na požadavcích projektu a Cisco IOS. [12, 13, 14]

## 4.2 QEMU

Rychlý emulátor procesoru, který využívá dynamický překlad a dosahuje velmi dobrého výkonu. Umí emulovat architektury x86, Sparc a ARM. Může běžet bez ovladače jádra hostitele, i přesto poskytuje dostatečný výkon. Lze jej také použít s virtuálním počítačem s KVM k spuštění virtuálních strojů skoro nativní rychlostí. Poskytuje různé modely hardwaru a zařízení, což mu umožňuje provozovat různé OS. Qemu má dva základní provozní režimy, a to plnou emulaci systému a emulaci v uživatelském režimu. V plném režimu je emulován celý systém včetně periférií a lze na něm provozovat různé operační systémy. Může být použit k virtualizaci více OS na jednom počítači. Druhý provozní režim se nazývá emulace v uživatelském režimu a můžeme na něm spustit programy, které byly zkompileovány jinou sadou instrukcí (např. spuštění aplikací určené pro jiný CPU). [15]

## 4.3 VirtualBox a VMware

Tyto programy poskytují snadný způsob virtualizace zařízení. Obě tyto pracovní stanice podporují virtualizaci VT-x pro CPU AMD i Intel. VirtualBox i VMware poskytují mnoho výhod. Mohou provozovat většinu operačních systémů, umožňují nám vytvářet libovolně velké topologie GNS3, které zahrnují počítače a servery, na nichž jsou spuštěny různé softwary. Musíme brát v potaz jejich nevýhody. Jsou náročnější na zatížení prostředků počítače. Pokud bychom chtěli pouze otestovat připojení v síti, je vhodnější použít VPCS či Docker Ubuntu. Pro využití robustnější sady síťových nástrojů jsou výhodnější právě tyto nástroje. Důležité je, že GNS3 integruje pouze VirtualBox a VMware a nemá kontrolu nad konfigurací virtuálních strojů (VM). Veškerá konfigurace VM je řízena virtualizačním softwarem, což znamená, že nemůžeme kopírovat nebo přesouvat svůj virtuální počítač jako součást projektu GNS3. [16, 17]

## 4.4 DOCKER

Využívá virtualizace na úrovni operačního systému a poskytuje software, který se nazývá kontejner. Kontejnery sdílejí jádro OS s ostatními kontejnery a jsou od sebe odděleny. Mají vlastní knihovny a konfigurační soubory. Díky podpoře Dockeru

v GNS3 máme možnost spouštět více kontejnerů jako součást vytvořené síťové topologie. Pokud chceme simulovat počítač či server, který poskytuje určitý typ služby a nechceme přitom významně zatížit paměť, tak je Docker ideální volbou. Oproti VirtualBoxu a Wmware využívá méně paměti a snižuje spotřebu CPU. V GNS3 je využíván k emulaci lehkého počítače s Linuxem, často se používá jako náhrada VPCS. [13, 14]

## 4.5 Wireshark

Wireshark je open source program a je k dispozici ke stažení zdarma. Jedná se o analyzátor síťových paketů. Analyzátor síťových paketů si můžeme představit jako měřící zařízení toho, co se děje uvnitř kabelového vedení. Pomocí něj lze zobrazit podrobné informace o zachycených paketech. Podporuje celou škálu protokolů a má uživatelsky přívětivé rozhraní. Pokud bychom chtěli zobrazit pouze určitý druh provozu, můžeme nastavit filtrování podle potřebných kritérií (např. zobrazení pouze UDP datových jednotek). Filtrování využívají správci sítě k řešení problémů se sítí.

Příklady Funkcí:

- Zachycení a uložení dat ze síťových rozhraní
- Zobrazení paketů a jeho podrobných informací
- Vyhledávání paketů
- Exportování dat v různých formátech
- Vytváření různých statistik a grafů

Dále je hojně využíván k prozkoumání bezpečnosti v síti, k určení interních síťových protokolů, k ladění implementací protokolů a k ověřování síťových aplikací. [18, 19]

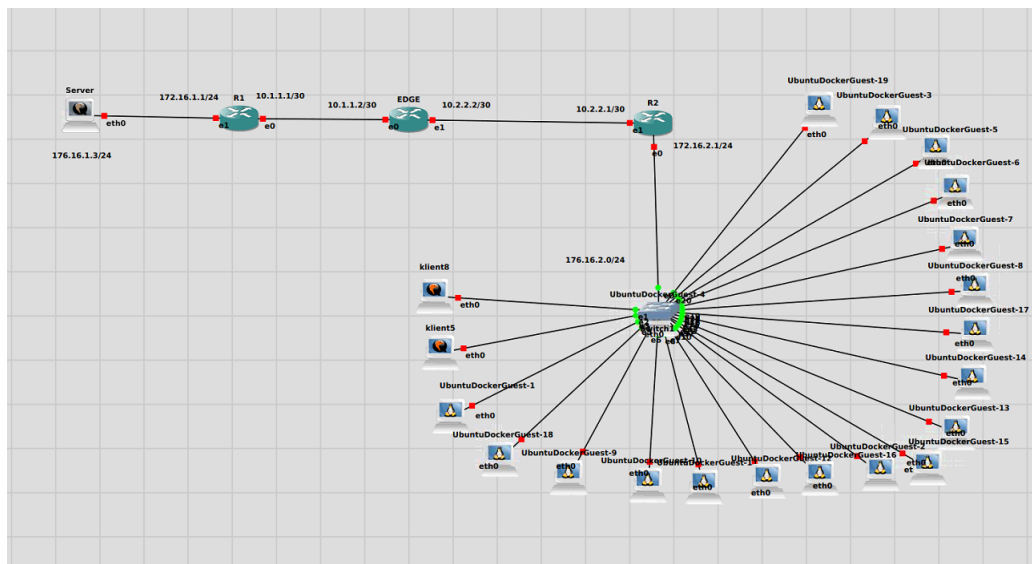
## 5 Simulační scénář

Tato kapitola pojednává o praktické části práce. V první části je znázorněn postup při vytváření topologie. V druhé části jsou popsány nástroje, které se používají pro měření a generování síťového provozu.

### 5.1 Postup při vytváření topologie

Hlavní topologie byla vytvořena v simulačním programu GNS3 a je znázorněna na obrázku 5.1 a pro multimediální přenos byla upravena do podoby, která je na obrázku 6.1. Pro linku nejpomalejší rychlosti pomocí programu Dynamips byly přidány tři Cisco routery c2691. Tyto routery obsahují adaptéry GT96100-FE. Pro linku Gigabit Ethernet a Fast Ethernet byly použity routery OpenWrt. Dále byla vytvořena topologie, na které probíhal provoz generovaný skriptem. Pro směrování danou sítí byly nakonfigurovány také routery OpenWrt, které jsou rozšiřitelnou distribucí GNU/Linux. Tato topologie obsahuje jeden server a celkem dvacet klientů. Pro připojení velkého množství koncových zařízení, které se nacházejí v podsíti, je použit EthernetSwitch. Tento Switch byl již součástí GNS3. Topologie pro simulaci generování provozu obsahuje celkový počet 25 zařízení v síti. Pro klienty byl použit adresní rozsah 172.16.2.x/24. Aby nemusela být u každého klienta staticky nastavena IP adresa, byl router R2 nakonfigurován jako DHCP server.

Obr. 5.1: Topologie sítě pro simulaci generování provozu



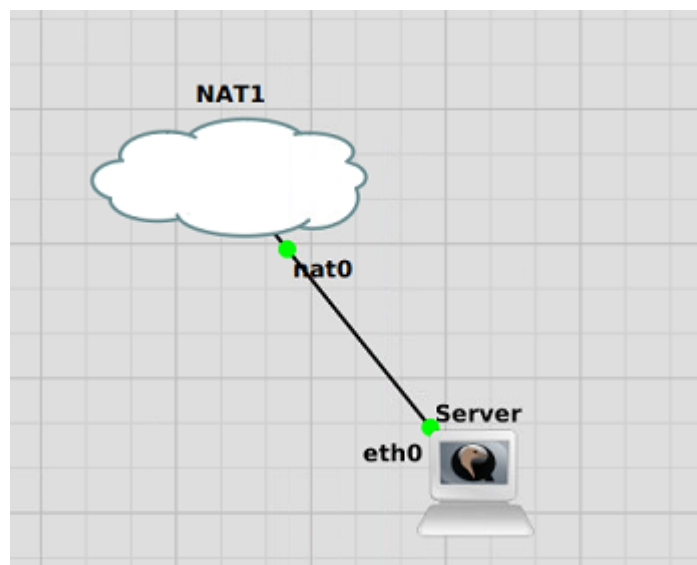
Pro dvě koncová zařízení byl použit QEMU VM, na kterém je nainstalován OS Linux Ubuntu. Pro další klienty byl použit DOCKER, jelikož mají pouze roli přijímače. Virtuální počítače QEMU narozdíl od DOCKER stanic obsahují grafické uživatelské prostředí. K uskutečnění simulace multimediálního přenosu byl aplikován software, který umožňuje multimediální přenos. Nazývá se VLC (VideoLAN Client). Server i koncové zařízení obsahují tento program a pomocí něj probíhá vysílání, resp. přijímání multimediálního obsahu. Součástí programu jsou streamovací protokoly.

Ve všech topologiích je komunikace mezi zařízeními zajištěna pomocí dynamického směrovacího protokolu OSPF. Příkazem Traceroute bylo zjištěno, jakou cestou jsou pakety směrovány. Pro ověření funkčnosti sítě byl použit příkaz Ping. Dotazování probíhalo přímo na konkrétní adresy v síti.

## 5.2 Nástroje pro měření a generování provozu

Pomocí NATu lze klienta připojit k internetu. Díky tomu si můžeme stáhnout a nainstalovat příslušné nástroje pro měření a generování provozu. Při připojení přes NAT není klient přímo dostupný z internetu nebo z LAN. Pokud bychom k připojení použili Cloud, tak by klient v síti byl normálně dostupný. Uzel NAT vyžaduje buď GNS3 VM, nebo počítač s OS Linux s nainstalovaným libvirt. Libvirt je nezbytný pro vytvoření rozhraní vibr0, jinak by NAT nefungoval. Ve výchozím nastavení má NAT nakonfigurovaný DHCP server s rozsahem 192.168.122.0/24. [12]

Obr. 5.2: Spojení přes NAT



### 5.2.1 Ostinato

Ostinato je open source program, generátor síťového provozu. Umožňuje vytvářet datové jednotky, nastavit jejich počet a přenosovou rychlost. Má přehledné grafické uživatelské rozhraní. V něm můžeme zobrazit statistiky v reálném čase, ve kterých jsou informace o vygenerovaném toku dat a informace o jednotlivých paketech. Je podporován jak na systému Windows, tak na OS Linux. Pomocí něj lze testovat zatížení sítě nebo její funkčnost.

Příklady jeho funkcí:

- Vytváření a konfigurace streamů
- Rychlost a počet paketů
- Podpora protokolů Ethernet, IPv4, IPv6, TCP, UDP, ICMP, IGMP, RTP
- Nastavení libovolné hodnoty pro jakékoli pole protokolu
- Změna IP či MAC adresy za provozu [20]

### 5.2.2 Síťové nástroje příkazového řádku

V Linuxu si můžeme nainstalovat různé síťové nástroje z výchozího úložiště operačního systému. K tomu slouží správce balíčků *apt*. V praktické části jich několik bylo použito.

**CBM** (Color Bandwidth Meter) je jednoduchý síťový nástroj, který umožňuje zobrazit aktuální síťový provoz na jednotlivých rozhraní. Jak již z názvu vyplývá, umožňuje zobrazit šířku pásma, odeslané bajty, přijaté bajty a také celkový počet bajtů. [21]

**Nload** je dalším síťovým nástrojem příkazového řádku, který sleduje provoz v reálném čase. Dále sleduje využití šířky pásma, minimální a maximální využití sítě (přijaté a odeslané bajty, průměrná hodnota). Můžeme si zvolit, jaké rozhraní chceme zobrazit. Provoz si lze zobrazit v grafickém provedení. [21]

**BMON** (Bandwidth Monitor Network) je jednoduchý, avšak dost výkonný nástroj pro monitorování sítě. Zachytává síťové statistiky a vizualizuje je v přehledném grafickém formátu. Lze tu také zobrazit přenosovou rychlost. [21]

**Iptraf** je uživatelsky přívětivý nástroj pro monitorování síťového provozu. Pomocí IPtraf si lze zobrazit přehledné statistiky jako, např. IP pakety, TCP a UDP datové položky. [21]

**Iperf** je open source program využíváný k měření propustnosti sítě v reálném čase. Pomocí testu IPerf UDP dokážeme změřit jitter i ztrátu paketu, pomocí testu Iperf TCP se měří šířka pásma. Podporuje jak IPv4, tak IPv6. Je spolehlivější oproti webovým měřicím aplikacím. Požadavkem měření je, aby program byl spuštěn na dvou počítačích. Je to nutné z hlediska porovnání výsledků. Jeden počítač je klient, druhý počítač je server. Klient požádá o měření, vytvoří se TCP spojení

pro přenos konfiguračních parametrů a výsledků. Poté se sestaví spojení pro měření parametrů a proběhne přenos od klienta k serveru. Po konci měření jsou data poslána klientovi a dojde k zobrazení na obou aplikacích. Pokud bychom chtěli přidat grafické rozhraní, je možná instalace balíčku Jperf, který je napsán v jazyce Java. V Jperf lze zobrazit grafy pro šířku pásma nebo pro jitter. [22, 23]

**Traceroute** je aplikace, která využívá nástroje pro diagnostiku sítě. Pomocí těchto nástrojů můžeme sledovat cestu paketu sítě (od zdroje k cíli). Poskytují informace o počtu skoků, u každého uzlu je zobrazena IP adresa a doba odezvy. Aplikace traceroute pracuje s uzly, které se nachází na síťové vrstvě. Tento nástroj specifikuje, kde se nachází problém v síti. Na rozdíl od aplikace Ping, která nás pouze informuje, že v síti existuje nějaký problém. Jeho nevýhoda je v tom, že nedokáže zobrazit historická data přenosu, proto může nastat situace, kdy firewall blokuje požadavky ICMP zprávy, a tím pádem traceroute bude ukazovat neúplná data. Při stanici S1 pošle žádost o odezvu cílové stanice S2, TTL se nastaví nejdříve na hodnotu 1. První router na trase paket zahodí a pošle ICMP zprávu o vypršení časovače (získáme RTT pro skok 1). Tím se S1 dozví adresu prvního routeru. Následně se znovu odešle žádost o odezvu cílové stanice S2, avšak teď se hodnota TTL zvýší na 2. Paket již projde prvním směrovačem a k jeho zahození dojde až na druhém routeru a pošle ICMP zprávu jako v případě prvního routeru (získáme RTT pro skok 2). Tento mechanismus je stejný v celé síti, dokud paket nedorazí do cílové stanice S2. [3, 24]

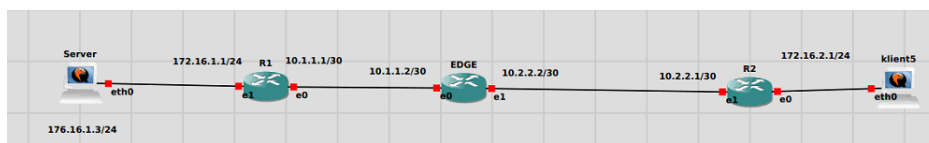
**Ntopng** je nástroj pro monitorování sítě, díky němuž mohou být zobrazeny statistiky a různé datové toky, které lze zachytit z provozu zachyceného serverem. Tento nástroj je schopen monitorovat veškeré pakety, které jsou posílány na jedno či více rozhraní. Obsahuje další funkce jako např. zobrazení propustnosti a provozu v reálném čase, nebo také zobrazení latence (RTT). Dále nabízí statistiky TCP (opakování přenosu, ztracené pakety). Je to open source program, který je k dispozici na GitHubu. Pro Ubuntu je instalační balíček dostupný v úložišti. [25]

## 6 Multimediální přenosy

Při realizaci simulačního scénáře gigabitové linky byla nalezena různá omezení. Omezení rychlosti bylo způsobené emulačním softwarem. To bylo vyřešeno přidáním virtuální síťové karty od firmy Intel model e1000 (Gigabit Ethernet). Serveru byla přidána čtyři virtuální jádra procesoru a paměť RAM o velikosti 2048 MB. Stanice, které přijímaly multimediální a datový přenos, měly nastaveny paměť RAM na 800 MB a obsahovaly dvě virtuální jádra procesoru. Kontrola správné přenosové rychlosti byla otestována při spojení dvou QEMU stanic přes přenosovou technologii Gigabit Ethernet. K testování byl použit Iperf. První stanice byla nastavena jako server, ta sloužila k naslouchání. Druhá stanice byla nastavena jako klient. Výsledná propustnost byla 995 Mbit/s. Pro zajištění gigabitové šířky pásma byl v prvním případě přidán mezi klienty router Mikrotik, který byl emulován programem QEMU. Rychlost byla omezena na 1 Mbit/s. K odstranění omezení bylo vyzkoušeno přihlášení a zaregistrování zkušební verze v konzoli Mikrotiku. Omezení se nepodařilo opravit, z tohoto důvodu bylo nutné vyměnit router. Byl využit OpenWrt router.

V této kapitole jsou charakterizovány multimediální přenosy, které probíhaly pomocí programu VLC. Tento typ přenosu se nazývá Over-The-Top (OTT). Videoobsah je přenášén po síti bez možnosti kontroly poskytovatelem internetových služeb. Pro streamování multimediálního obsahu byla využita část topologie znázorněná na obrázku 6.1. Multimediální streaming nejdříve běžel na lince 1000BASE-T, na které může být teoreticky šířka pásma až 1000 Mbit/s. Pro simulaci Fast Ethernetu byla síťová karta změněna u všech zařízení z e1000 na kartu i82551. Pro srovnání výsledků byla použita linka 100BASE-TX. Tato linka umožňuje až 100Mbit/s. Před začátkem měření byla zjištěna skutečná propustnost linky také pomocí programu Iperf. Skutečná přenosová rychlost se rovnala 980 Mbit/s pro Gigabit Ethernet a pro pomalejší technologii 87 Mbit/s. Každá vrstva TCP/IP modelu má různou režii, a proto nebylo možné dosáhnout teoretické propustnosti. Také proběhlo měření na zastaralejší lince (10BASE-T), která umožňovala maximální teoretickou propustnost 10 Mbit/s. Cisco routery byly v režimu VendorLock. Výrobce se tímto chrání proti zneužití v reálném nasazení. Reálná propustnost tedy byla 1 Mbit/s. Měření streamu probíhalo po dobu jedné minuty a pro srovnání kodeků byla stanovena doba měření 30 sekund. K měření byla použita videa různých kvalit. Jejich parametry jsou popsány v tabulkách níže.

Obr. 6.1: Linka 1000BASE-T a 100BASE-TX



Tab. 6.1: Parametry pro video 360p

| Video 360p                 |                 |
|----------------------------|-----------------|
| Použitý kodek              | MPEG-4          |
| Trvání videa               | 299 sekund      |
| Datová rychlost            | 650 kb/s        |
| Bitová rychlost            | 778 kb/s        |
| Zvuková přenosová rychlost | 128 kb/s        |
| FPS                        | 25              |
| Celkový počet rámců        | 7475            |
| Rozlišení videa            | 640x360         |
| Velikost videa             | 29 378 082 B    |
| Barevná hloubka            | 0,14 bitů/pixel |

Tab. 6.2: Parametry pro video 480p

| Video 480p                 |                 |
|----------------------------|-----------------|
| Použitý kodek              | H.264           |
| Trvání videa               | 307 sekund      |
| Datová rychlost            | 809 kb/s        |
| Bitová rychlost            | 934 kb/s        |
| Zvuková přenosová rychlost | 125 kb/s        |
| FPS                        | 60              |
| Celkový počet rámců        | 18 420          |
| Rozlišení videa            | 854x480         |
| Velikost videa             | 36 712 598 B    |
| Barevná hloubka            | 0,04 bitů/pixel |



Tab. 6.3: Parametry pro video 720p

| Video 720p                 |                 |
|----------------------------|-----------------|
| Použitý kodek              | MPEG-4          |
| Trvání videa               | 232 sekund      |
| Datová rychlost            | 3176 kb/s       |
| Bitová rychlost            | 3305 kb/s       |
| Zvuková přenosová rychlost | 128 kb/s        |
| FPS                        | 30              |
| Celkový počet rámců        | 6960            |
| Rozlišení videa            | 1280x720        |
| Velikost videa             | 96 067 578 B    |
| Barevná hloubka            | 0,12 bitů/pixel |

Tab. 6.4: Parametry pro video 1080p

| Video 1080p                |                 |
|----------------------------|-----------------|
| Použitý kodek              | MPEG-4          |
| Trvání videa               | 324 sekund      |
| Datová rychlost            | 11 788 kb/s     |
| Bitová rychlost            | 11 916 kb/s     |
| Zvuková přenosová rychlost | 128 kb/s        |
| FPS                        | 25              |
| Celkový počet rámců        | 8100            |
| Rozlišení videa            | 1920x1080       |
| Velikost videa             | 509 530 668 B   |
| Barevná hloubka            | 0,24 bitů/pixel |

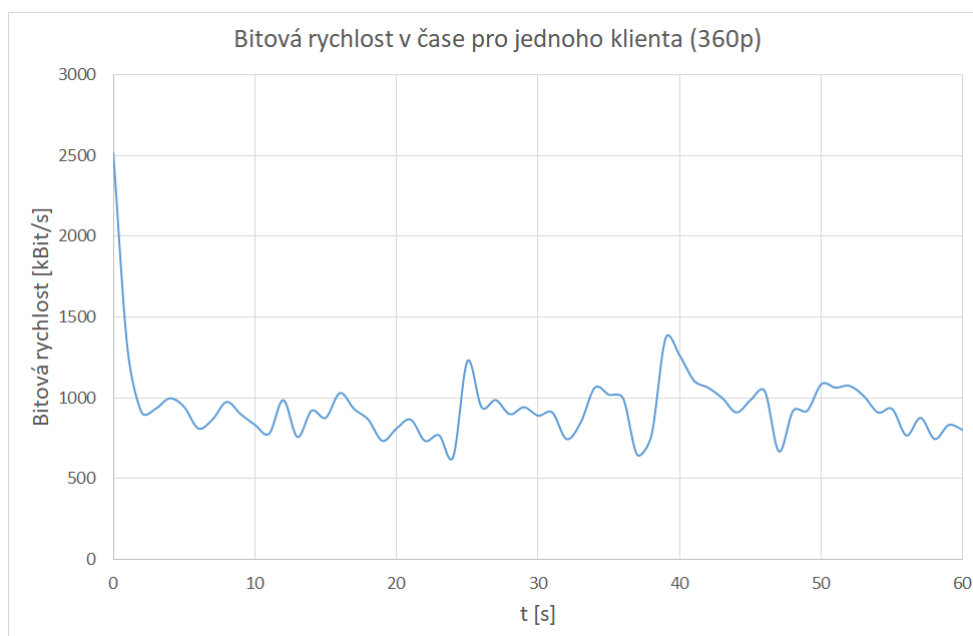
## 6.1 RTP stream na lince 1000BASE-T

V této kapitole se nacházejí výsledky multimediálního přenosu. Tento přenos probíhal na lince s největší propustností. Vysoká propustnost linky umožňuje přenos videoobsahu bez ztráty na kvalitě obrazu.

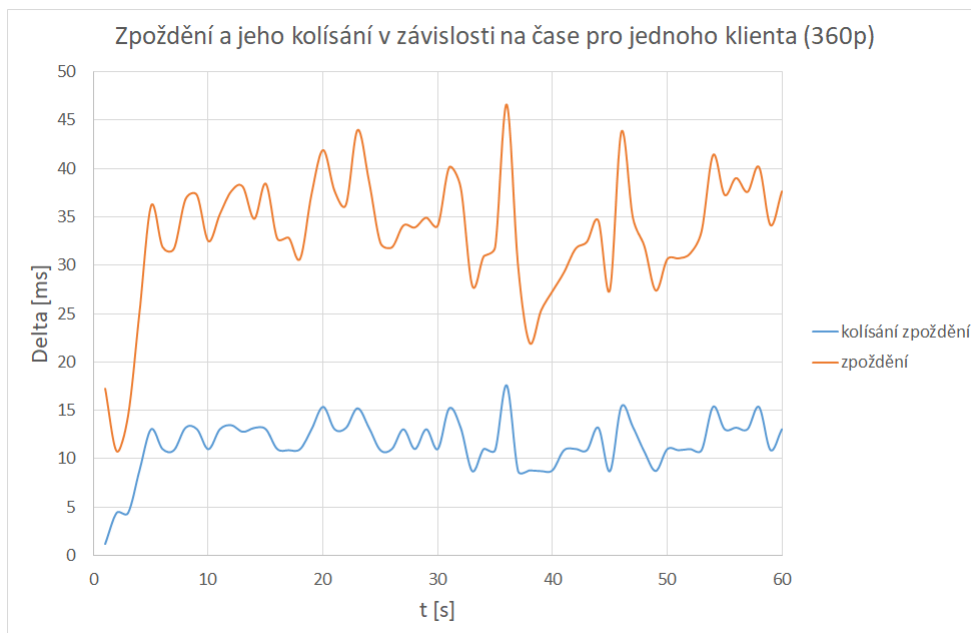
### 6.1.1 RTP stream video 360p

Pro tuto simulaci bylo vybráno video s parametry definované v tabulce 6.1. V tomto případě byla průměrná rychlost určena na hodnotě 949,46 kBit/s. Na začátku streamu byla změřena nejvyšší přenosová rychlost a ta dosáhla 2517,55 kBit/s. Nejnížší přenosová rychlost v jednom okamžiku klesla na hodnotu 635,68 kBit/s. Kolísání zpoždění přenosu průměrně dosahovalo hodnoty 11,56 ms a průměrná hodnota zpoždění 33,30 ms. Průběh zpoždění v čase neměl výrazné výkyvy v hodnotách. Nejvyšší hodnota kolísání zpoždění byla 17,60 ms a nejnižší hodnota 1,13 ms. Maximální zpoždění činilo 46,60 ms a nejnižší hodnota vykazovala 10,78 ms. Při přenosu vybraného videa nejnižší kvality nedošlo k žádné ztrátovosti. Výsledné grafy jsou zobrazeny na obrázcích 6.2 a 6.3.

Obr. 6.2: Bitová rychlost v čase (360p)



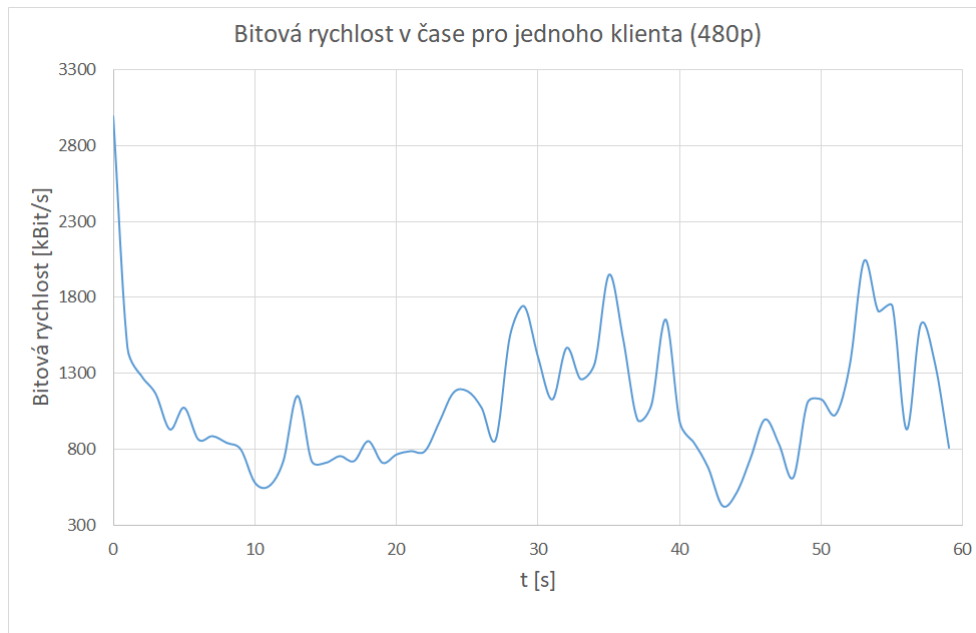
Obr. 6.3: Zpoždění a jeho kolísání v závislosti na čase (360p)



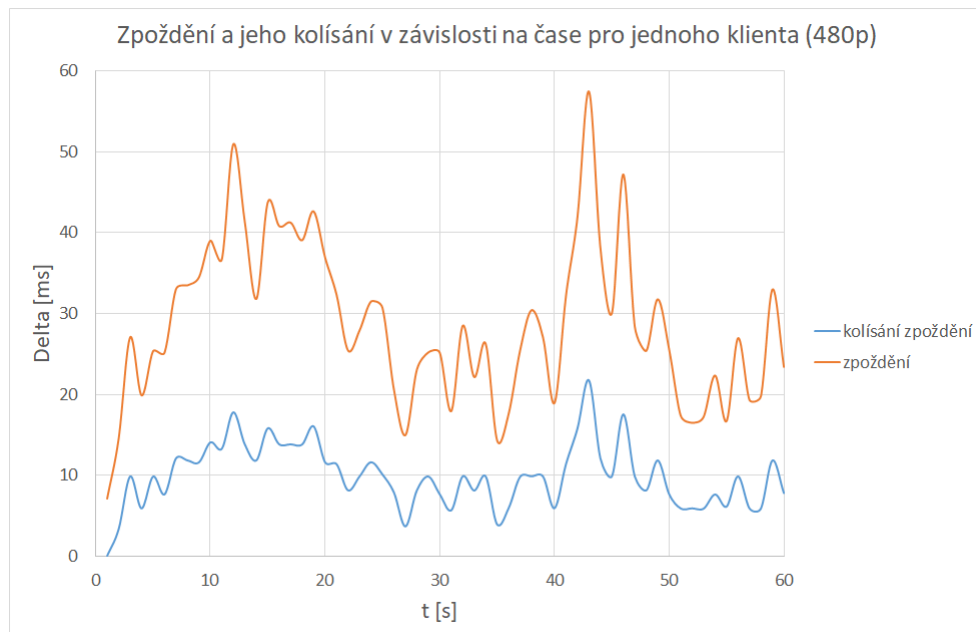
### 6.1.2 RTP stream video 480p

Vybrané video střední kvality 480p má parametry určené v tabulce 6.2. Při multi-mediálním přenosu bylo dosaženo průměrné bitové rychlosti 1096,8 kBit/s. Nejvyšší bitová rychlost byla 2992 kBit/s, naopak nejnižší byla pouhých 429,3 kBit/s. Počet přenesených kBit v čase je znázorněn na grafu 6.4. Na grafu 6.5 je vynesena grafická závislost zpoždění a jeho kolísání v čase. Z tohoto grafu vyplývá, že průměrné kolísání zpoždění se pohybovalo okolo hodnoty 9,93 ms. Průměrná hodnota zpoždění nabývala hodnoty 28,64 ms. Nejvyšší hodnota zpoždění byla v druhé polovině přenosu (57,47 ms). V této části nejvíce kolísalo zpoždění (21,77 ms). Ani v tomto měření nebyla žádná procentuální ztrátovost paketů.

Obr. 6.4: Bitová rychlost v čase (480p)



Obr. 6.5: Zpoždění a jeho kolísání v závislosti na čase (480p)

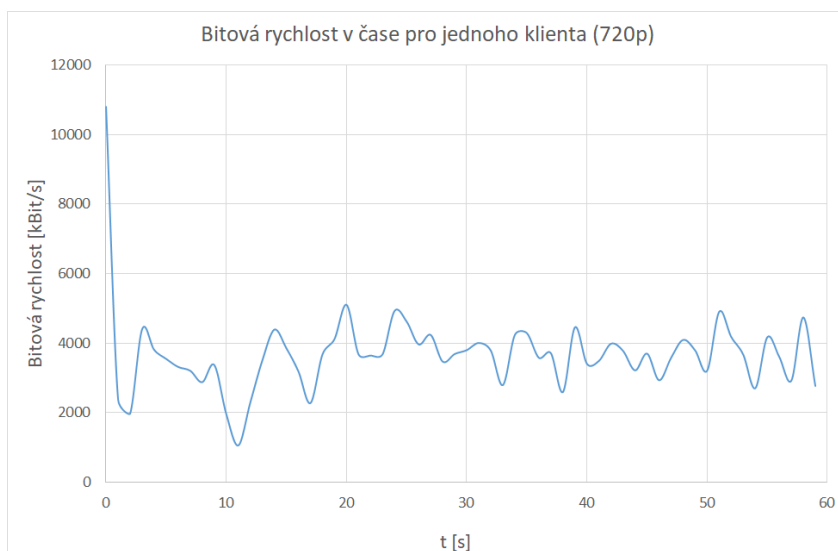


### 6.1.3 RTP stream video 720p

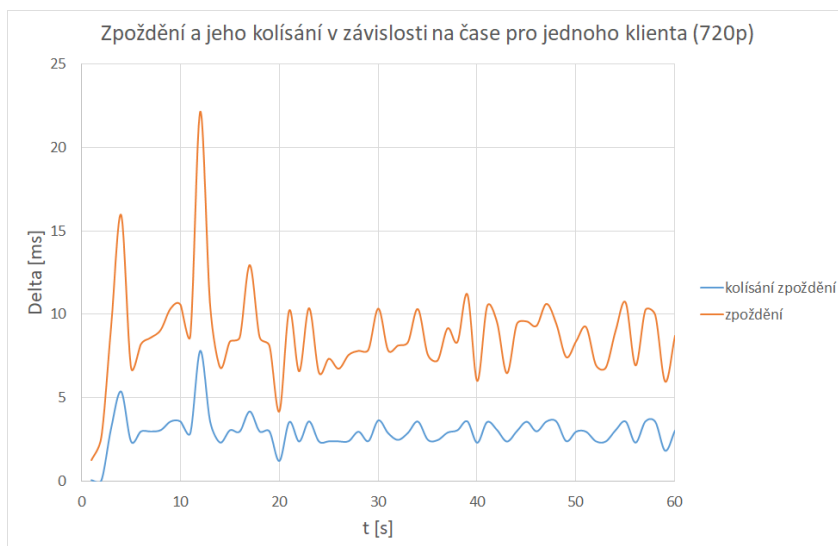
HD video a jeho parametry jsou k nalezení v tabulce 6.3. Průběh bitové rychlosti v čase je zobrazen na grafu 6.6. Průměrná bitová rychlost činila 3678,7 kBit/s. Na začátku streamu byla zaznamenána nejvyšší přenosová rychlost 10795,6 kBit/s

(10,7 Mbit/s). Nejnižší přenosová rychlost byla určena v excelu pomocí funkce MIN na 1052,16 kBit/s. Závislost zpoždění a jeho kolísání je zobrazena na obrázku 6.7. V této závislosti se nachází jedna špička zpoždění (22,16 ms), která je způsobena poklesem bitové rychlosti. V tuto chvíli také výrazně kolísalo zpoždění (7,83 ms). V rámci celého přenosu se zpoždění pohybovalo průměrně kolem hodnoty 8,73 ms. Ztrátovost tohoto přenosu se rovnala 0,47 % (10 paketů).

Obr. 6.6: Bitová rychlost v čase (720p)



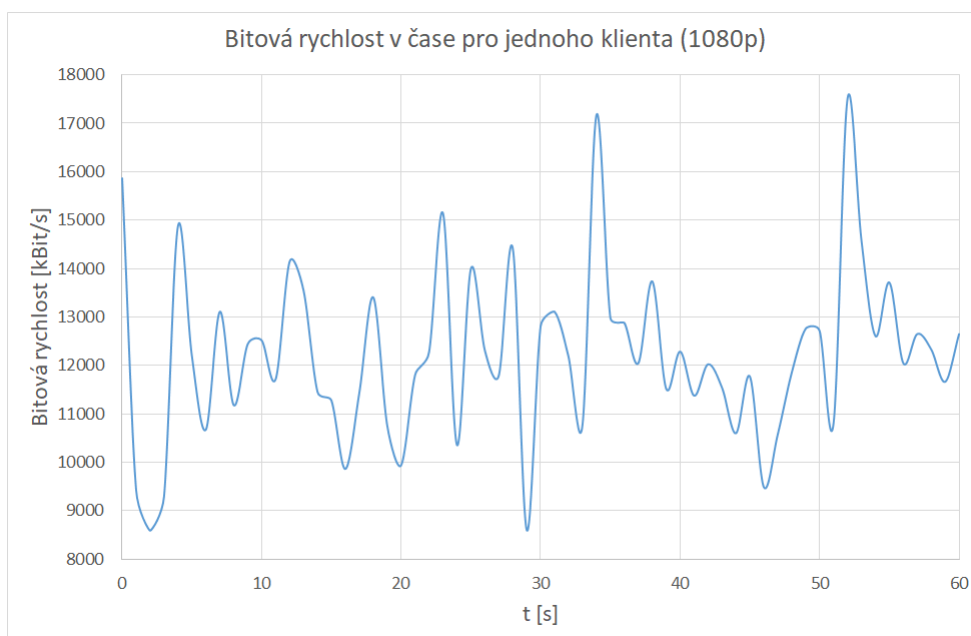
Obr. 6.7: Zpoždění a jeho kolísání v závislosti na čase (720p)



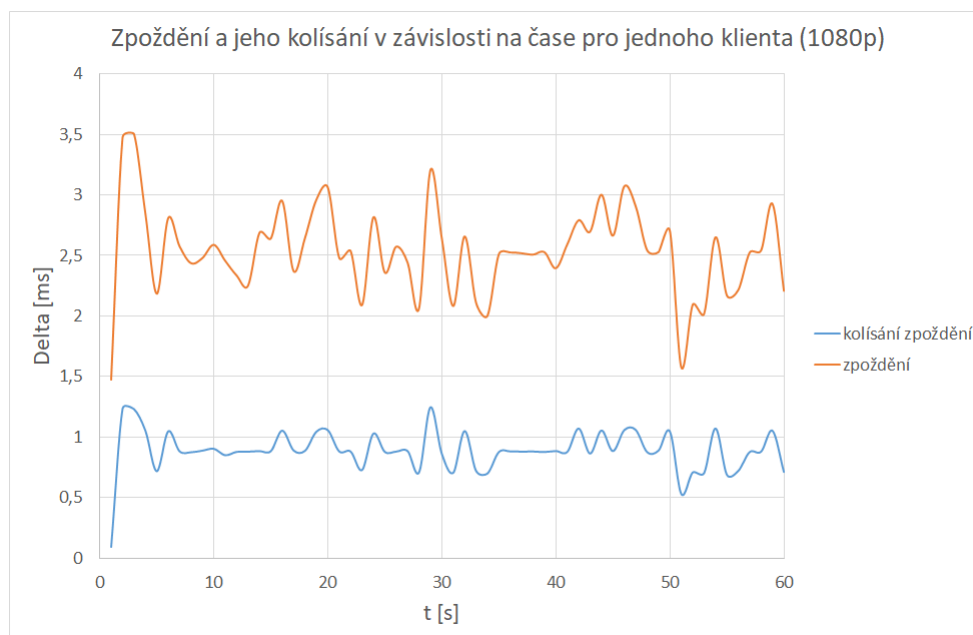
### 6.1.4 RTP stream video 1080p

Full HD video má parametry, které se nachází v tabulce 6.4. Na grafu 6.8 lze pozorovat, že průměrná bitová rychlost činila 12214,59 kBit/s. Výrazné špičky bitové rychlosti v průběhu přenosu byly způsobeny rychlejšími scénami videa, a tedy vyšším datovým tokem. Největší hodnota rychlosti byla 17503,1 kBit/s. Poklesy v přenosové rychlosti symbolizovaly méně náročné scény. Minimální hodnota bitové rychlosti byla 8592,64 kBit/s. Graf 6.9 znázorňuje průběh zpoždění a jeho kolísání v čase. Průměrná hodnota zpoždění byla změřena a ukazovala hodnotu 2,53 ms. Pouze chvílemi se pohybovala přes 3 ms. Největší zpoždění bylo 3,50 ms. V tomto měření byla zaznamenána největší procentuální ztrátovost 5,06 % (3632 paketů). Výsledná ztrátovost nezpůsobovala výrazné problémy v rozlišení či funkčnosti videa.

Obr. 6.8: Bitová rychlost v čase (1080p)



Obr. 6.9: Zpoždění a jeho kolísání v závislosti na čase (1080p)



### 6.1.5 Souhrnné porovnání

Největší bitová rychlost byla změřena pro Full HD video. Nejmenší průměrná bitová rychlost byla zjištěna u videa nejnižší kvality. Největší hodnota zpoždění byla zaznamenána pro video 480p, avšak průměrně nejvyšší hodnotu mělo video 360p. Ze změřených údajů vyplývá, že čím vyšší byla bitová rychlost, tím více klesala hodnota zpoždění. Naopak pokud klesala hodnota rychlosti, tím větší bylo zpoždění přenosu.

Tab. 6.5: Změřené parametry pro přenos videa 360p

| Změřené parametry pro přenos videa 360p |                          |               |
|---|--------------------------|---------------|
| Hodnota                                 | Bitová rychlost [kBit/s] | Zpoždění [ms] |
| Průměrná                                | 949,46                   | 33,30         |
| Max.                                    | 2517,55                  | 46,60         |
| Min.                                    | 635,68                   | 10,78         |

Tab. 6.6: Změřené parametry pro přenos videa 480p

| <b>Změřené parametry pro přenos videa 480p</b> |                                 |                      |
|--|---------------------------------|----------------------|
| <b>Hodnota</b>                                 | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná                                       | 1096,83                         | 28,64                |
| Max.   | 2992                            | 57,47                |
| Min.   | 429,36                          | 7,05                 |

Tab. 6.7: Změřené parametry pro přenos videa 720p

| <b>Změřené parametry pro přenos videa 720p</b> |                                 |                      |
|--|---------------------------------|----------------------|
| <b>Hodnota</b>                                 | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná                                       | 3678,74                         | 8,73                 |
| Max.   | 10795,6                         | 22,16                |
| Min.   | 1052,16                         | 1,28                 |

Tab. 6.8: Změřené parametry pro přenos videa 1080p

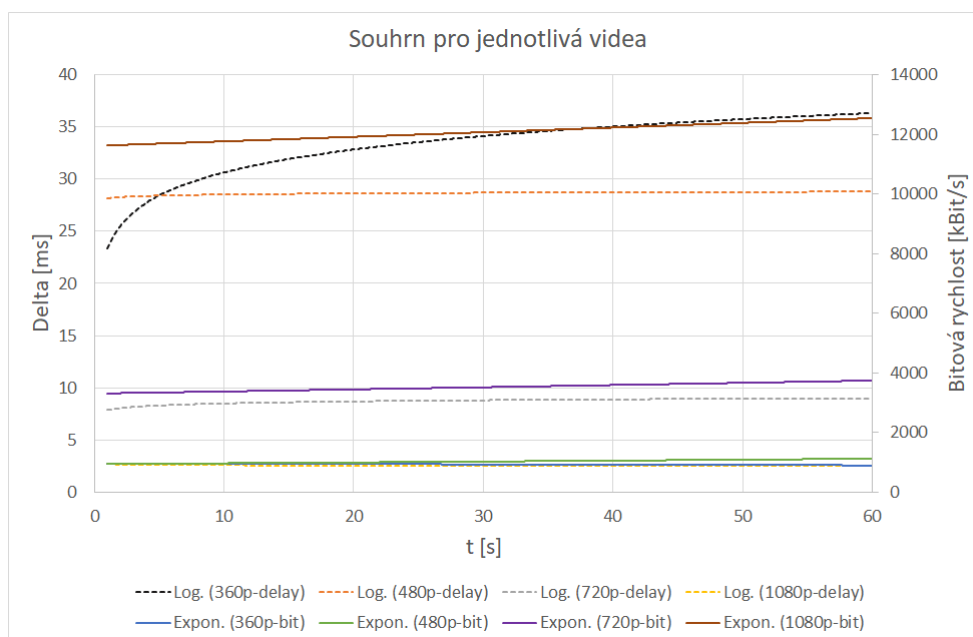
| <b>Změřené parametry pro přenos videa 1080p</b> |                                 |                      |
|---|---------------------------------|----------------------|
| <b>Hodnota</b>                                  | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná  | 12214,59                        | 2,53                 |
| Max.  | 17503,1                         | 3,50                 |
| Min.  | 8592,64                         | 1,47                 |

Pro přehledné porovnání jednotlivých veličin jsou řady grafu zobrazeny pomocí spojnice trendu. V souhrnném grafu 6.10 lze pozorovat nepřímou úměrnost mezi bitovou rychlostí a zpožděním. Čím větší byla bitová rychlost, tím menší zpoždění nastávalo během multimediálního přenosu. Největší přenosovou rychlost vykazovalo video 1080p a zároveň mělo nejnižší zpoždění v průběhu přenosu. Pro video HD kvality (720p) byla zaznamenána bitová rychlost, která v čase mírně exponenciálně rostla. Jeho zpoždění nabývalo vyšších hodnot, než tomu bylo u videa 1080p. Pro video 480p zpoždění bylo poměrně konstantní a bitová rychlost byla téměř stejná jako pro video 360p. Nejvyšší logaritmický růst zpoždění nastal pro video nejnižší kvality, naopak zde byla zaznamenána nejnižší bitová rychlost.

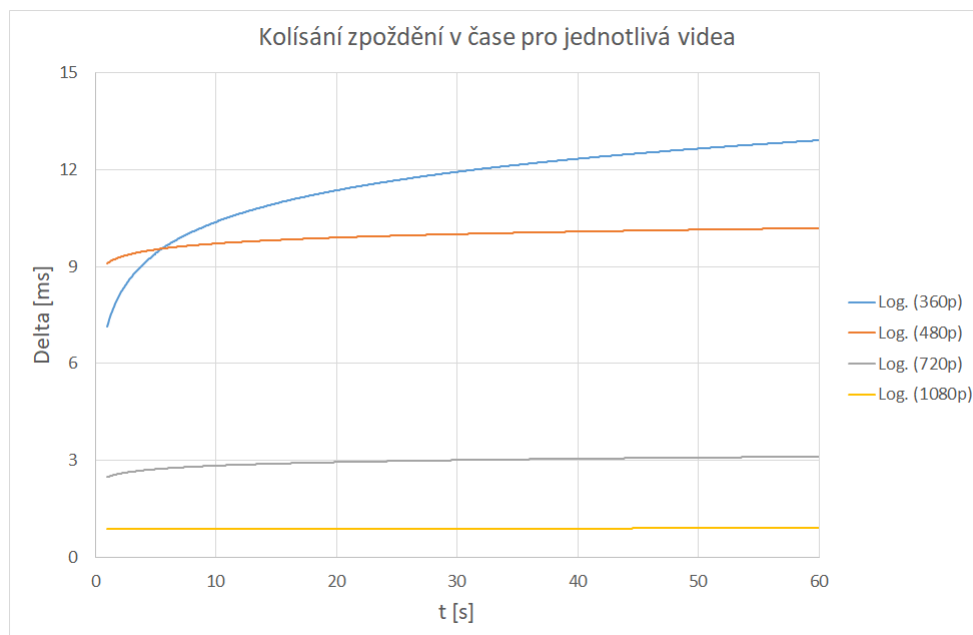
Kolísání zpoždění v závislosti na čase je znázorněno na obrázku 6.11. Zpoždění kolísalo nejvíce pro video 360p (logaritmicky). Nejmenší hodnoty kolísání zpoždění nastaly pro Full HD video.



Obr. 6.10: Souhrnný graf pro jednotlivá videa



Obr. 6.11: Souhrnný graf pro kolísání zpoždění v závislosti na čase pro jednotlivá videa



### 6.1.6 Vývoj zpoždění v čase

V tabulkách 6.9 a 6.10 jsou zaznamenány rovnice, do kterých byly dosazeny (za neznámou  $x$ ) hodnoty času. Tyto hodnoty se pohybovaly od 3600 sekund (1 hodina) do 86 400 sekund (24 hodin). Pro Full HD video se zpoždění snížilo o 0,16 ms. Pro

HD video se zpoždění zvýšilo z 10,14 ms na 11,04 ms. Oproti předchozímu videu se jednalo o nárůst zpoždění o 81 %. Pro video 480p zpoždění nepatrně vzrostlo z hodnoty 29,41 ms na 29,89 ms. V porovnání s HD videem došlo k nárůstu zpoždění o 63 %. Nejvyšší nárůst zpoždění byl vypočítán pro video nejnižší kvality. Zpoždění se zvýšilo o 10,1 ms. Při porovnání videí 480p a 360p zpoždění vzrostlo o 50 %. Nejmenší zpoždění bylo vypočítáno pro video 1080p a nejvyšší zpoždění bylo u videa 360p. Procentuální rozdíl zpoždění mezi těmito dvěma videi činil 96 %.

Tab. 6.9: Výpočet logaritmické spojnice trendu pro 1080p a 720p

| <b>Výpočet rovnice spojnice trendu</b> |                              |                              |
|--|------------------------------|------------------------------|
| <b>Rovnice</b>                         | $y = -0,049 \ln(x) + 2,6916$ | $y = 0,2805 \ln(x) + 7,8526$ |
| <b>t [s]</b>                           | 1080p                        | 720p                         |
| 3600                                   | 2,29035423290223             | 10,1495272994066             |
| 14 400                                 | 2,22242580920736             | 10,5383828677007             |
| 28 800                                 | 2,18846159735992             | 10,7328106518478             |
| 43 200                                 | 2,16859380706262             | 10,8465436146721             |
| 57 600                                 | 2,15449738551249             | 10,9272384359949             |
| 72 000                                 | 2,14356335149809             | 10,9898302021385             |
| 86 400                                 | 2,13462959521519             | 11,0409713988192             |

Tab. 6.10: Výpočet logaritmické spojnice trendu pro 480p a 360p

| <b>Výpočet rovnice spojnice trendu</b> |                              |                              |
|--|------------------------------|------------------------------|
| <b>Rovnice</b>                         | $y = 0,1515 \ln(x) + 28,172$ | $y = 3,1763 \ln(x) + 23,321$ |
| <b>t [s]</b>                           | 480p                         | 360p                         |
| 3600                                   | 29,4125864023533             | 49,3307332659721             |
| 14 400                                 | 29,622609998063              | 53,7340200451972             |
| 28 800                                 | 29,7276217959178             | 55,9356634348098             |
| 43 200                                 | 29,78904975979625            | 57,2235422576938             |
| 57 600                                 | 29,8326335937726             | 58,1373068244223             |
| 72 000                                 | 29,8664398417967             | 58,8460776864617             |
| 86 400                                 | 29,894061557651              | 59,4251856473063             |

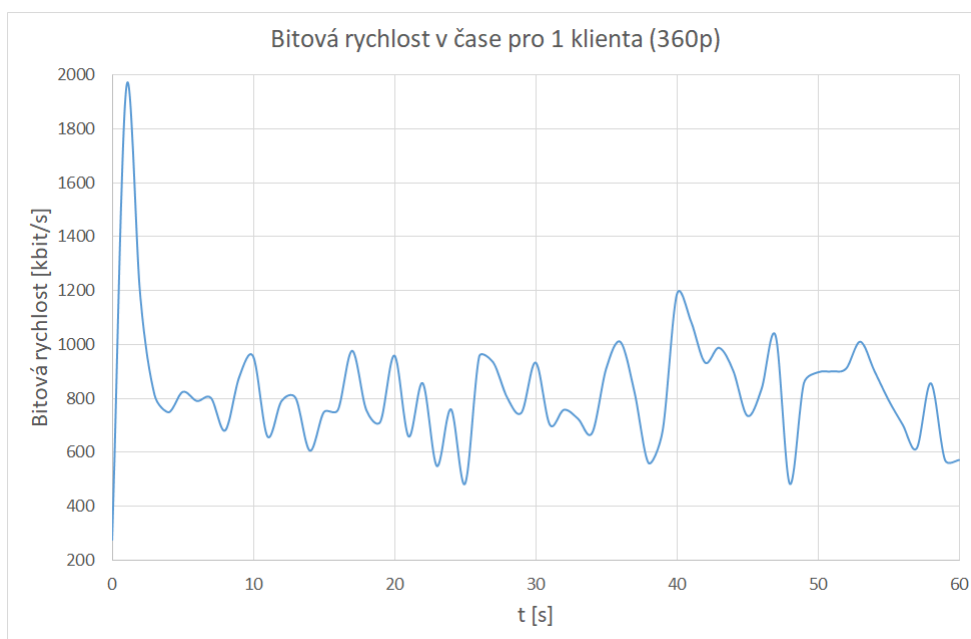
## 6.2 RTP stream na lince 100BASE-TX

K porovnání výsledků na gigabitové lince byla použita linka Fast Ethernet, která umožňuje teoretickou propustnost až 100 Mbit/s, tudíž zde byla 10x menší propustnost než v předchozím měření.

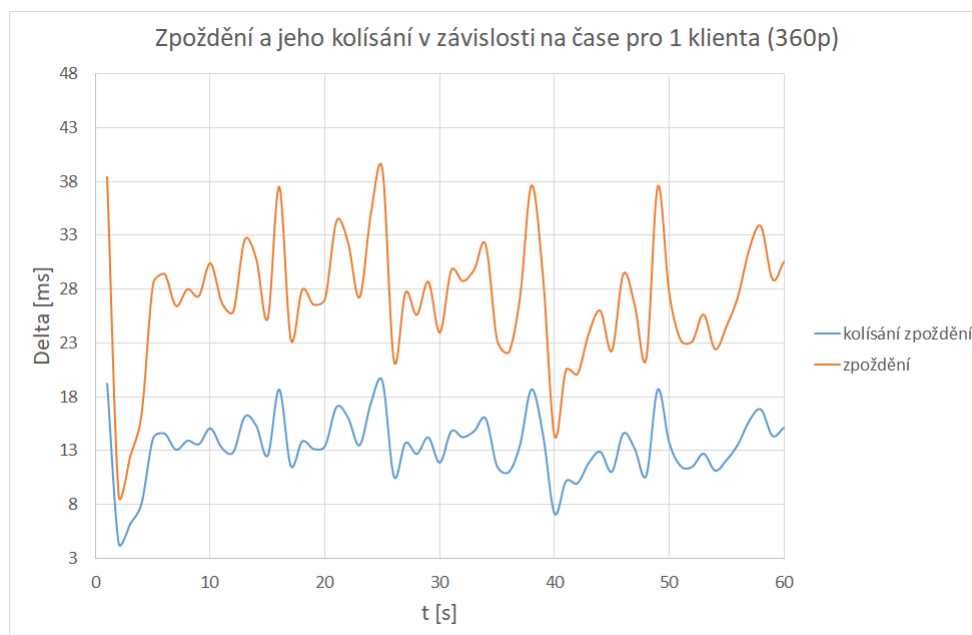
### 6.2.1 RTP stream pro video 360p

V rámci celého přenosu byla průměrná bitová rychlost 817,53 kBit/s. Na grafu 6.12 se nachází špička, která ukazuje nejvyšší rychlost (1951,58 kBit/s). Na konci přenosu při vypnutí streamu byla zaznamenána nejnižší rychlost 274 kBit/s. Zpoždění se průměrně pohybovalo okolo 27,10 ms a kolísalo při 13,47 ms. Na grafu 6.13 bylo k vidění několik špiček, ta nejvyšší představuje hodnotu 39,07 ms, kterou doplňuje nejvyšší hodnota kolísání 19,45 ms. Minimální hodnota zpoždění a jeho kolísání činilo 8,91 ms resp. 4,40 ms. Oproti přenosu na gigabitové lince zde byla zaznamenána ztrátovost 3 %.

Obr. 6.12: Bitová rychlost v čase (360p) - Fast Ethernet



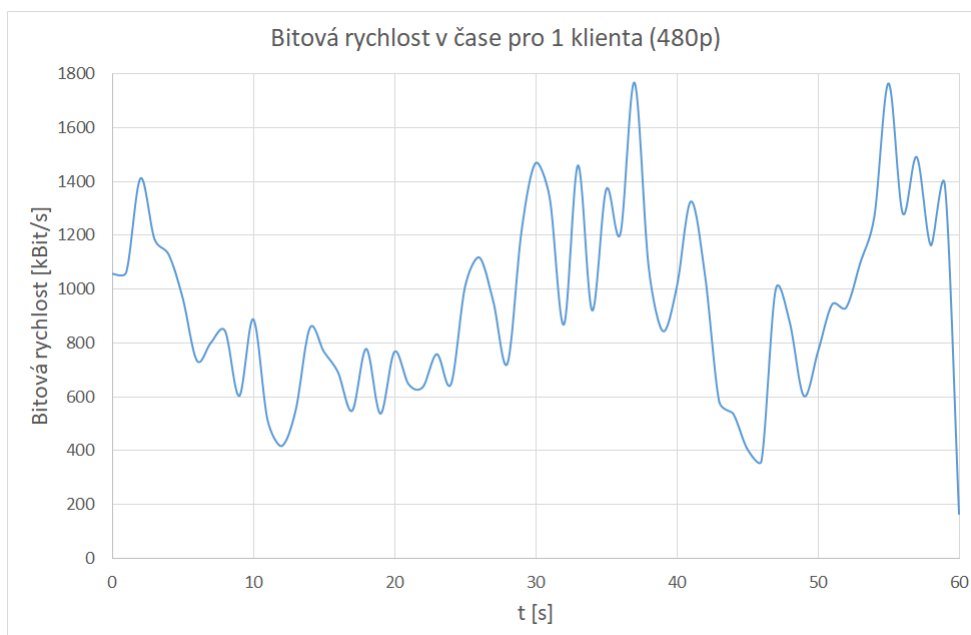
Obr. 6.13: Zpoždění a jeho kolísání v závislosti na čase (360p) - Fast Ethernet



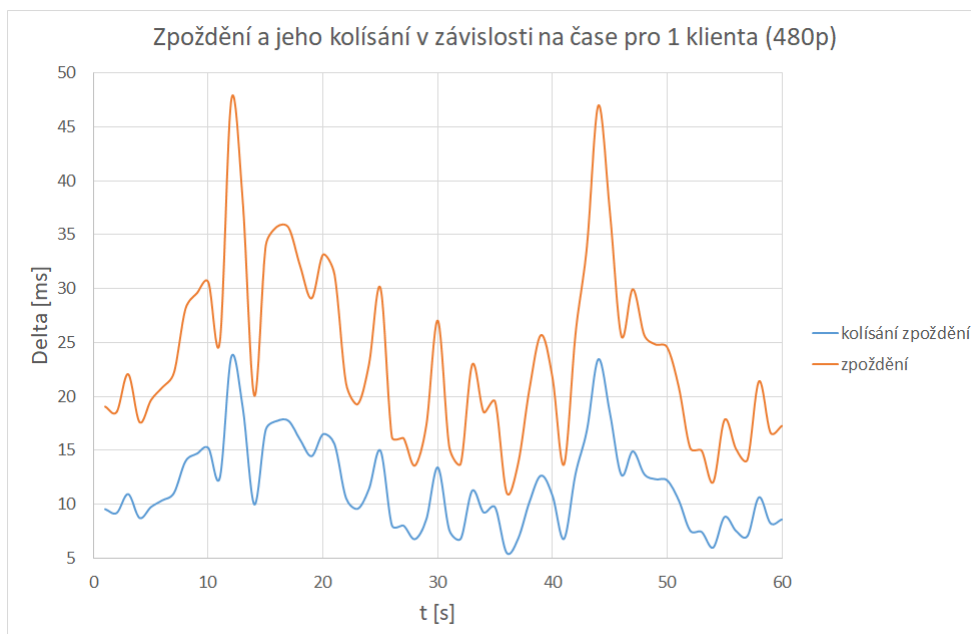
### 6.2.2 RTP stream pro video 480p

Streamované video 854x480 dosahovalo těchto výsledků. Stream videa se pohyboval v průměrné bitové rychlosti 936,81 kBit/s. Zpoždění klesalo v případě, že nastal značný nárůst bitové rychlosti (rychlejší scéna videa). Při nejvyšší bitové rychlosti 1767,16 kBit/s zpoždění kleslo na 11,03 ms, což značí nejmenší zpoždění při tomto streamu. V daný okamžik byla největší hodnota zpoždění 47,55 ms při bitové rychlosti 362,6 kBit/s. Tato hodnota nebyla nejmenší. Minimální rychlost (164,4 kBit/s) byla změřena na konci streamu, kdy docházelo k jeho ukončení. Pro tuto simulaci se ztrátovost rovnala 4,5 %.

Obr. 6.14: Bitová rychlost v čase (480p) - Fast Ethernet



Obr. 6.15: Zpoždění a jeho kolísání v závislosti na čase (480p) - Fast Ethernet

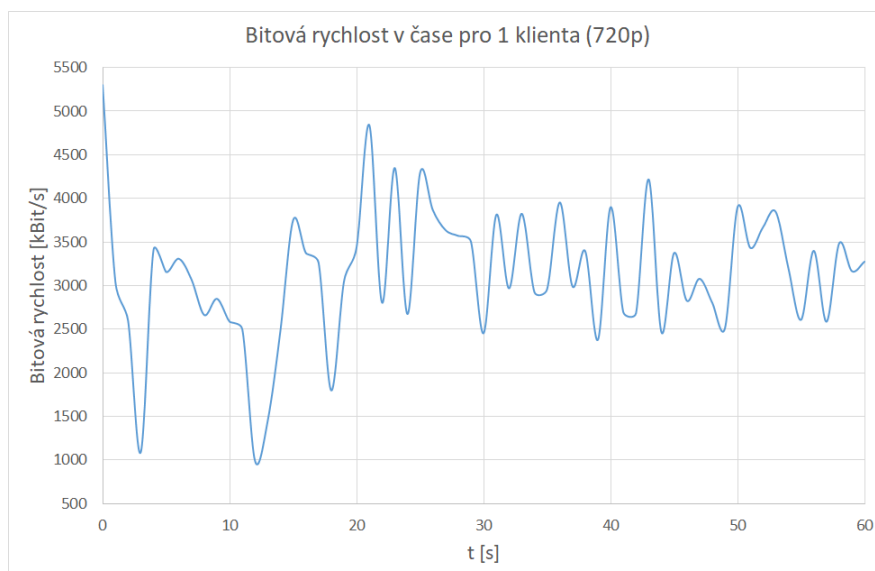


### 6.2.3 RTP stream pro video 720p

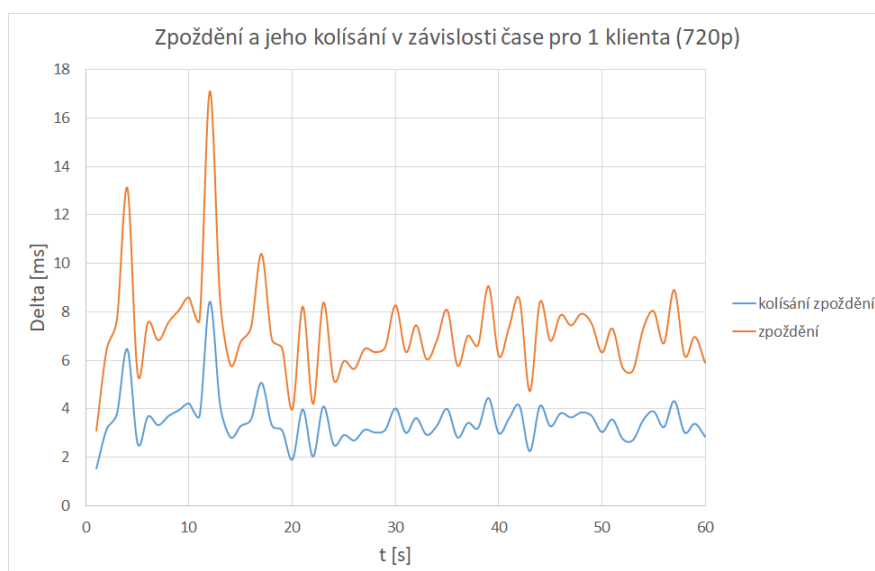
Průběh bitové rychlosti v čase je znázorněn na grafu 6.16. Začátek streamu doprovázela maximální bitová rychlost (5304,64 kBit/s). Průměrně se tato veličina pohybovala kolem 3140,66 kBit/s. Při málo náročné scéně (nižší datový tok) klesla

na minimální hodnotu 986,4 kBit/s. V tento okamžik (graf 6.17) bylo zpoždění maximální (17,12 ms) s maximálním kolísáním (8,42 ms). Průměrné zpoždění bylo okolo 7,20 ms. Oproti předešlému měření znovu došlo k nárůstu ztrátovosti na 13,44 % (2707 paketů). Tato ztrátovost způsobila občasné výpadky videa a nekvalitní obraz.

Obr. 6.16: Bitová rychlost v čase (720p) - Fast Ethernet



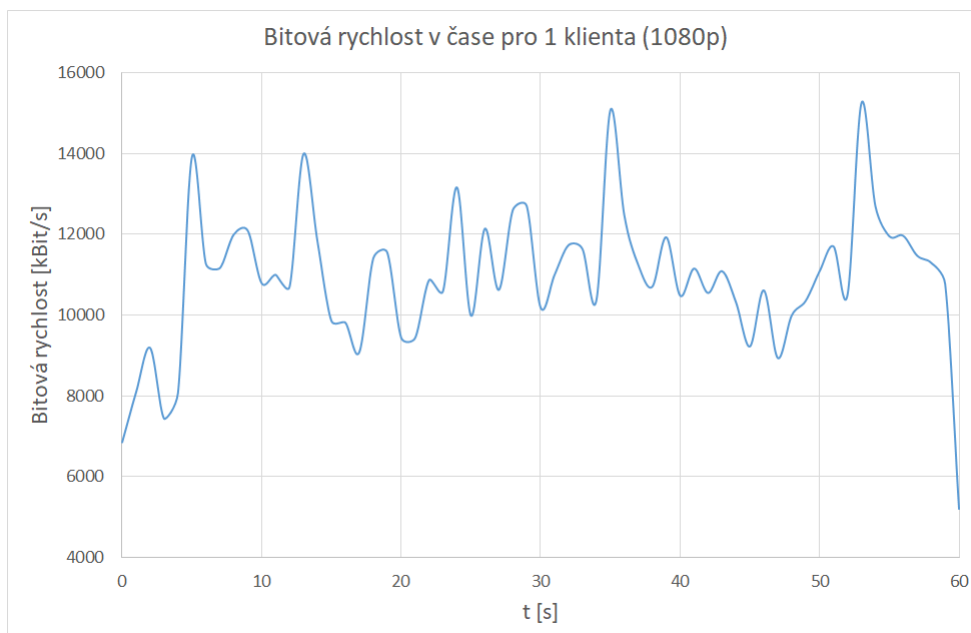
Obr. 6.17: Zpoždění a jeho kolísání v závislosti na čase (720p) - Fast Ethernet



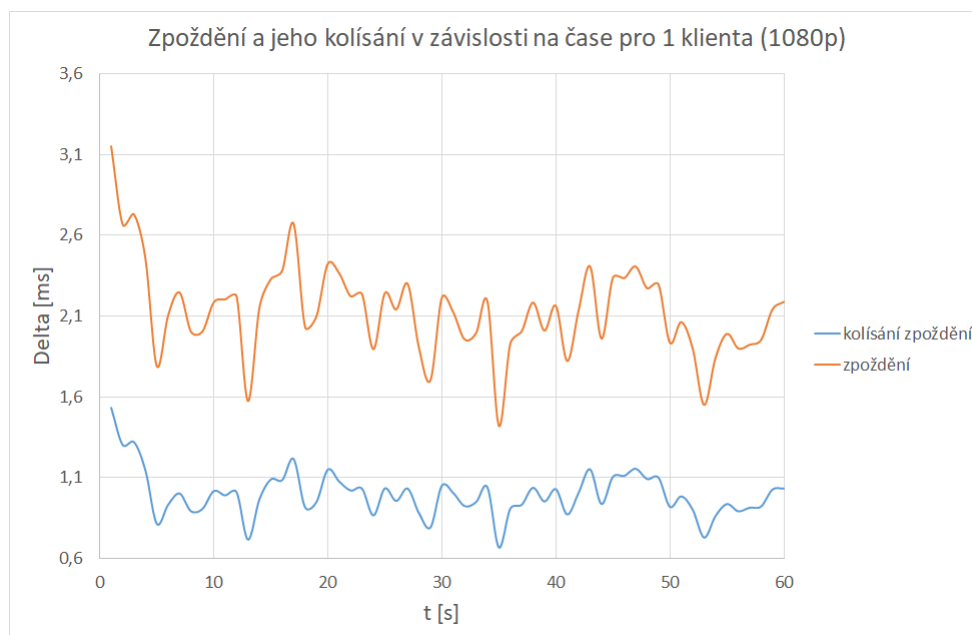
## 6.2.4 RTP stream pro video 1080p

Video o rozlišení 1920x1080 mělo následující průběh bitové rychlosti - viz graf 6.18. Na grafu je viditelná vysoká bitová rychlost (průměrně 10900,48 kBit/s). V posledních 10 sekundách se nacházela nejvyšší hodnota bitové rychlosti 15258,2 kBit/s. Výsledky průběhu zpoždění a jeho kolísání jsou ilustrovány na grafu 6.19. Průměrná hodnota zpoždění se dostala na hodnotu 2,13 ms. Maximální hodnota byla zaznamenána na 3,14 ms. Zpoždění kolísalo nejvíce při hodnotě 1,53 ms. U Full HD videa ztrátovost činila 13,29 % (9295 paketů), což je procentuálně nižší hodnota než pro HD video, avšak bylo zahozeno více paketů. Celková ztrátovost způsobila snížení rozlišení přenosu videa a měla negativní vliv na jeho funkčnost.

Obr. 6.18: Bitová rychlost v čase (1080p) - Fast Ethernet



Obr. 6.19: Zpoždění a jeho kolísání v závislosti na čase (1080p) - Fast Ethernet



## 6.2.5 Souhrnné porovnání

V této podkapitole jsou vypsány a porovnány změřené hodnoty pro jednotlivá videa. Nejvyšší hodnoty bitové rychlosti byly zaznamenány pro video 1080p. Během celého měření došlo k maximální hodnotě zpoždění 47,55 ms, které bylo změřeno pro video 480p. Průměrné zpoždění bylo největší pro video 360p. Ve srovnání s gigabitovou linkou došlo ke snížení průměrných hodnot zpoždění a bitové rychlosti u všech videí. Při těchto přenosech došlo k navýšení ztrátovosti. Procentuální ztrátovost ovlivnila výslednou kvalitu videa.

Tab. 6.11: Změřené parametry video 360p - Fast Ethernet

| Změřené parametry pro přenos videa 360p |                          |               |
|---|--------------------------|---------------|
| Hodnota                                 | Bitová rychlost [kBit/s] | Zpoždění [ms] |
| Průměrná                                | 817,53                   | 27,10         |
| Max.                                    | 1951,58                  | 39,07         |
| Min.                                    | 274                      | 8,91          |



Tab. 6.12: Změřené parametry video 480p - Fast Ethernet

| <b>Změřené parametry pro přenos videa 480p</b> |                                 |                      |
|--|---------------------------------|----------------------|
| <b>Hodnota</b>                                 | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná                                       | 936,81                          | 23,48                |
| Max.   | 1767,16                         | 47,55                |
| Min.   | 164,4                           | 11,03                |

Tab. 6.13: Změřené parametry video 720p - Fast Ethernet

| <b>Změřené parametry pro přenos videa 720p</b> |                                 |                      |
|--|---------------------------------|----------------------|
| <b>Hodnota</b>                                 | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná                                       | 3140,66                         | 7,20                 |
| Max.   | 5304,64                         | 17,12                |
| Min.   | 986,4                           | 3,11                 |

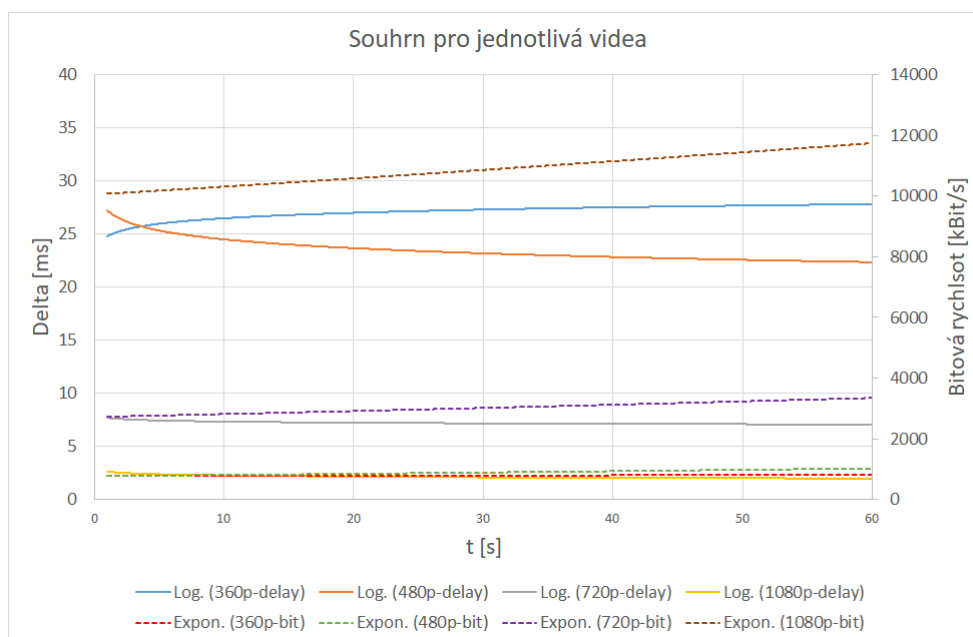
Tab. 6.14: Změřené parametry video 1080p - Fast Ethernet

| <b>Změřené parametry pro přenos videa 1080p</b> |                                 |                      |
|---|---------------------------------|----------------------|
| <b>Hodnota</b>                                  | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná  | 10900,48                        | 2,13                 |
| Max.  | 15258,2                         | 3,14                 |
| Min.  | 5206                            | 1,42                 |

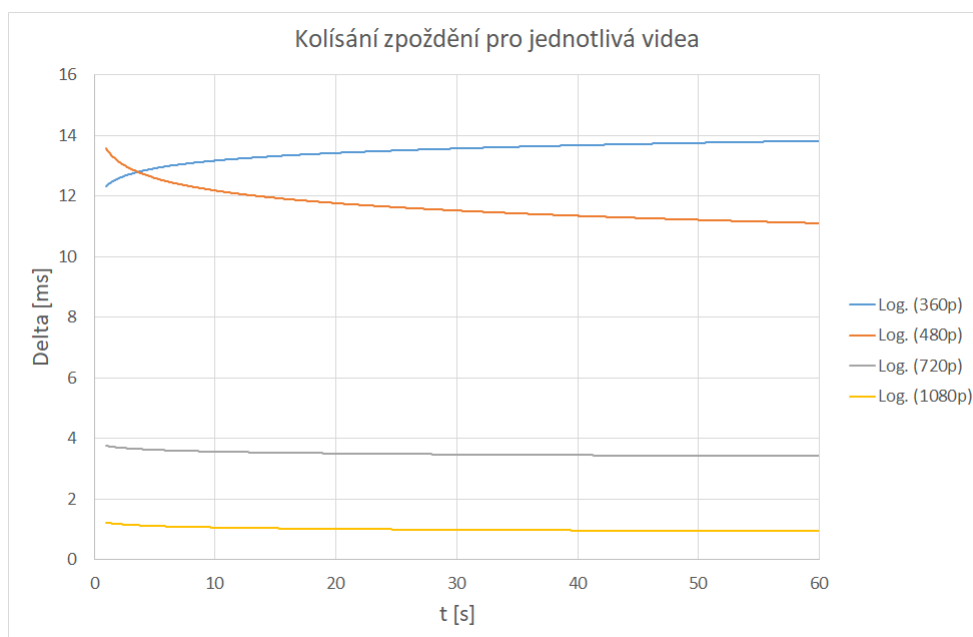
Pro názorné shrnutí byla bitová rychlost vynesena jako exponenciální spojnice trendu a zpoždění bylo zobrazeno pomocí logaritmické spojnice trendu. Pro video 1080p rychlost s časem exponenciálně rostla. Hodnoty Expon.(1080p) byly 3x větší než hodnoty pro Expon.(720p). Expon.(360p) a Expon.(480p) měly podobné průběhy, ale ke konci měření byla nepatrně vyšší rychlost pro video 480p. Nejmenší zpoždění bylo naměřeno pro 1080p. Pro video 720p se zpoždění pohybovalo v rozmezí 7-8 ms. Pro video 480p zpoždění logaritmicky klesalo v rozmezí 27-23 ms. Pro video nejnižší kvality zpoždění logaritmicky rostlo v rozmezí 25-28 ms.

Na obrázku 6.21 jsou vyneseny logaritmické spojnice trendu pro kolísání zpoždění. Kolísání zpoždění Log. (1080p) se pohybovalo pod hranicí 2 ms. Kolísání zpoždění pro Log.(720p) na počátku mírně rostlo, ale drželo se pod hranicí 4 ms. Log.(480p) měl klesající průběh. Log. (360p) rostl v závislosti na čase přibližně od hodnoty 12 ms až po 14 ms.

Obr. 6.20: Souhrnný graf pro jednotlivá videa - Fast Ethernet



Obr. 6.21: Souhrnný graf pro kolísání zpoždění v závislosti na čase - Fast Ethernet



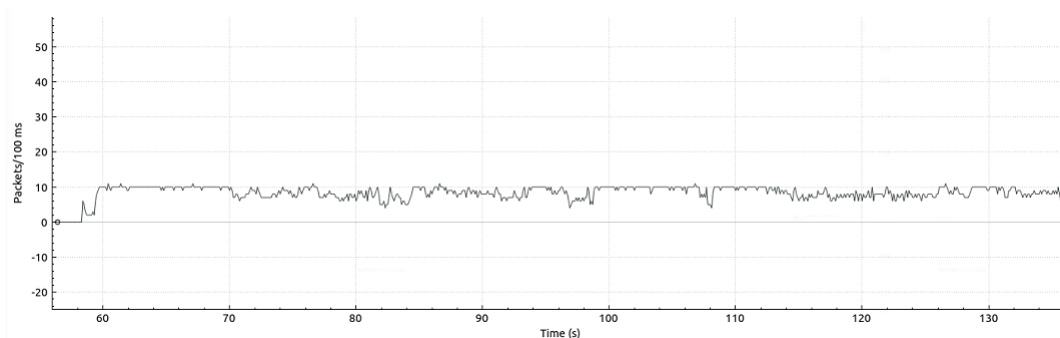
### 6.3 RTP stream na lince 10BASE-T

Tato linka s nejnižší propustností byla vybrána z důvodu otestování, jaká videa lze po ní přenést, aniž by byla významně degradována kvalita obrazu.

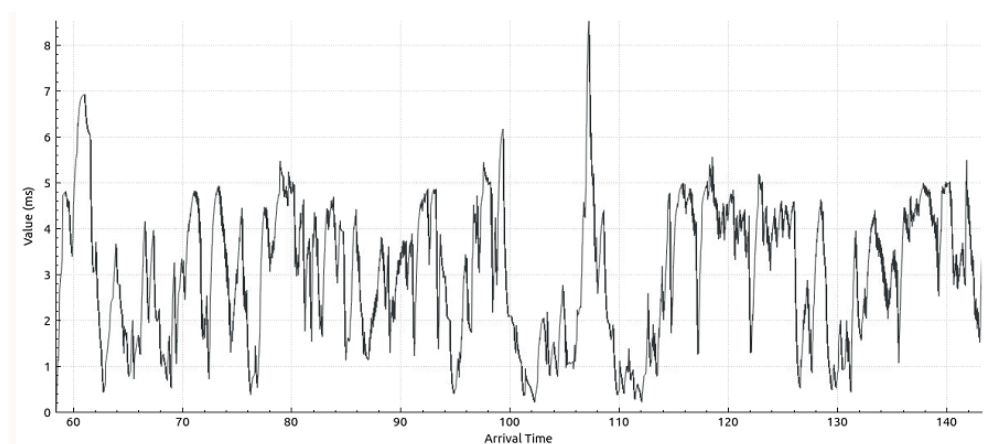
### 6.3.1 RTP stream video 360p

Pro streamování videa s rozlišením 360p byly dosaženy tyto výsledky. Při spuštění streamu byla patrná změna průběhu. Během přenosu nastalo mírné kolísání v počtu přenesených paketů. Celkový počet přenesených RTP paketů byl 4948, celkově 6778 kB. Průměrná rychlost odesílání byla 0,133 Mbit/s. Na obrázku 6.23 je znázorněno zpoždění doručení paketů, které se pohybovalo v normálních hodnotách (5 ms) a díky tomu neovlivňovalo přenos. Obraz byl plynulý a téměř bez přerušování. Ztrátovost paketů byla minimální a pohybovala se okolo 1 %. Pomocí Wiresharku byla také zobrazena průměrná velikost paketu. Průměrná hodnota délky paketu je 1370 B. Jsou zde viděny i pakety menších délek, jedná se o pakety ostatních protokolů (například OSPF).

Obr. 6.22: Počet přenesených paketů v čase pro video rozlišení 360p

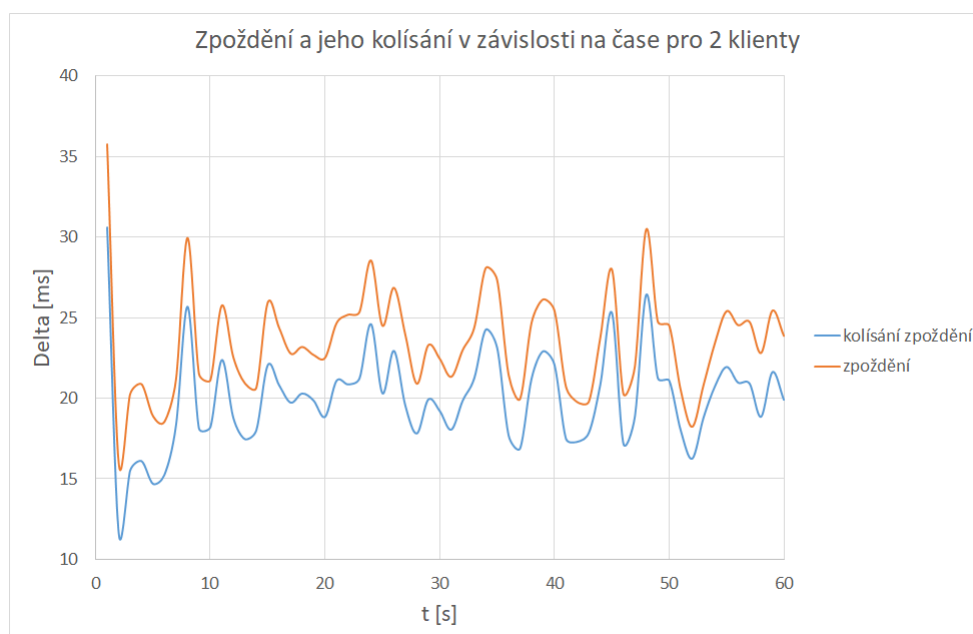


Obr. 6.23: Kolísání zpoždění pro video rozlišení 360p



Na základě předchozích výsledků a jejich zpracování byl proveden test přenosu ke dvěma klientům. Na grafu v této podkapitole je zobrazena závislost zpoždění a jeho kolísání na čase. Simulace probíhala po dobu jedné minuty. Během prvních sekund přenosu byla viditelná špička, která byla způsobena navazováním spojení a samotnou přípravou síťového přenosu. Zpoždění dosahovalo v určitých časech 30 ms a největší hodnota kolísání zpoždění se rovnala 31 ms. Pro prvního klienta byla zjištěna vysoká ztrátovost 41,49 % (2161 paketů) a druhého klienta byla zaznamenána ztrátovost 39,44 % (2126 paketů). Při tomto přenosu byla síť zatížena až příliš. Vlivem nedostatečné propustnosti linky nebylo možné uskutečnit stream ke dvěma klientům. V takovém případě jednou z možností je využití multicastového vysílání. Po síti by byl inicializován pouze jeden přenos na multicastovou skupinu a tím by docházelo k ušetření části šířky pásma.

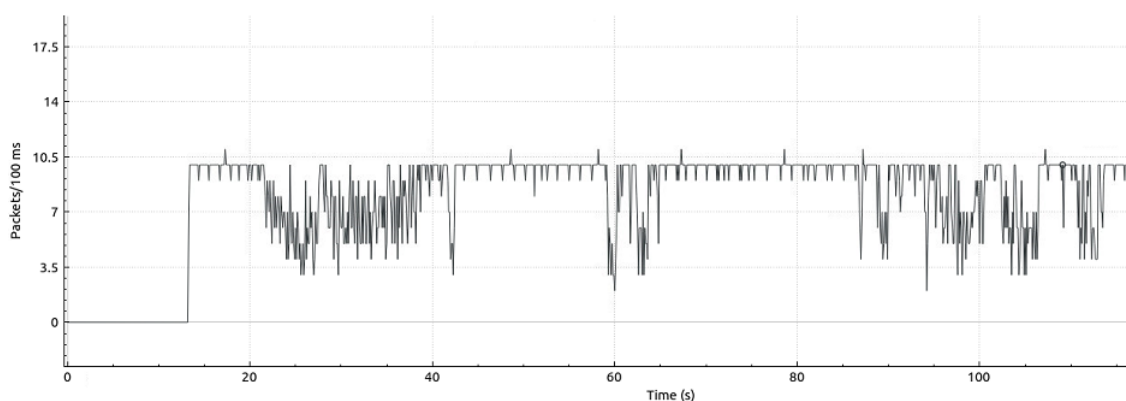
Obr. 6.24: Závislost zpoždění a jeho kolísání na čase pro 2 klienty (video 360p)



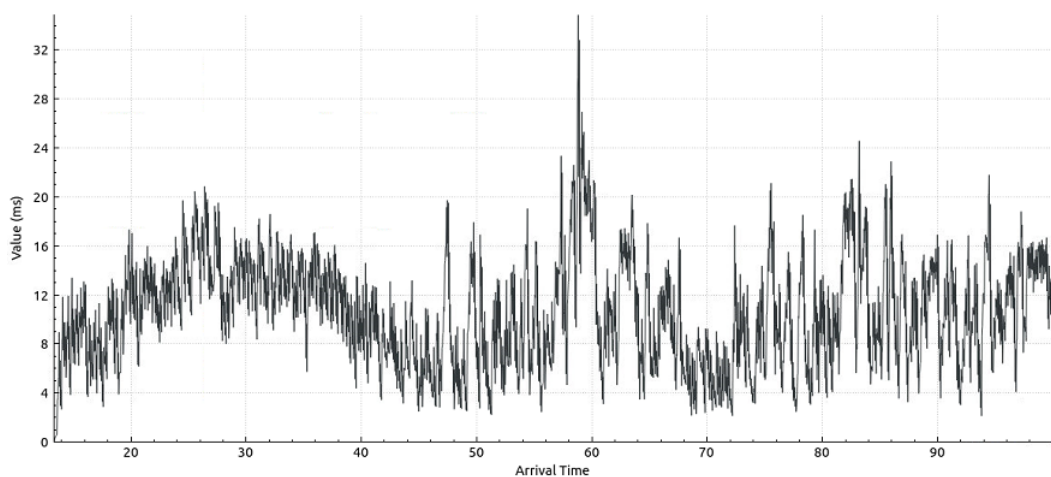
### 6.3.2 RTP stream video 480p

Pro streamování videa s rozlišením 480p byly dosaženy tyto výsledky. Ztrátovost paketů se pohybuje okolo 3,6 %. Tato hodnota nebyla uživatelsky rozpoznatelná. V průběhu docházelo ke zpoždění doručování paketů. Kolísání zpoždění bylo poměrně v přijatelných hodnotách, avšak byla zaznamenána přenosová špička (peak), která znázorňovala zaseknutí obrazu. Množství RTP paketů během jedné minuty se rovnalo hodnotě 5359, což odpovídá 7 341 kB při nejvyšší přenosové rychlosti 1,09 Mbit/s.

Obr. 6.25: Počet přenesených paketů v čase pro video rozlišení 480p

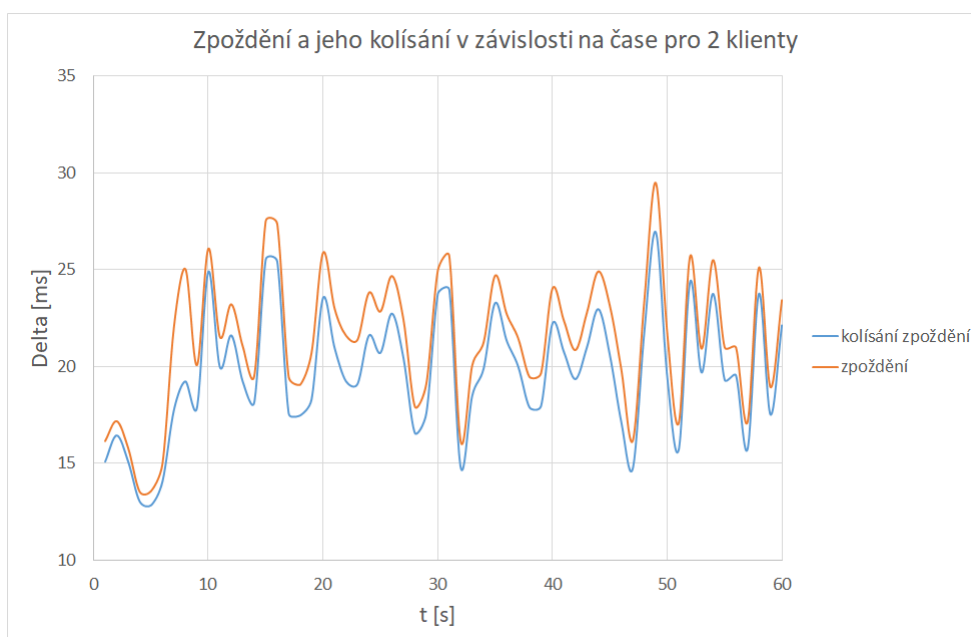


Obr. 6.26: Kolísání zpoždění pro video rozlišení 480p



Stejně jako v předchozí podkapitole byl zde odsimulován stream pro dva klienty. Na začátku přenosu lze vidět nejnižší hodnoty zpoždění a jeho kolísání z důvodu pozdějšího spuštění druhého streamu. Výrazné špičky ve zpoždění a jejich výrazné kolísání značily výpadek obrazu. Nejvyšší hodnota zpoždění byla 29 ms. Nekvalitní obraz dokazovala i vysoká ztrátovost pro oba klienty (2816 zahozených paketů ztrátovost 46,39 % a 2747 zahozených paketů se ztrátovostí 47,87 %). Z výsledků je zřejmé, že docházelo k výraznému ovlivňování obou multimediálních přenosů.

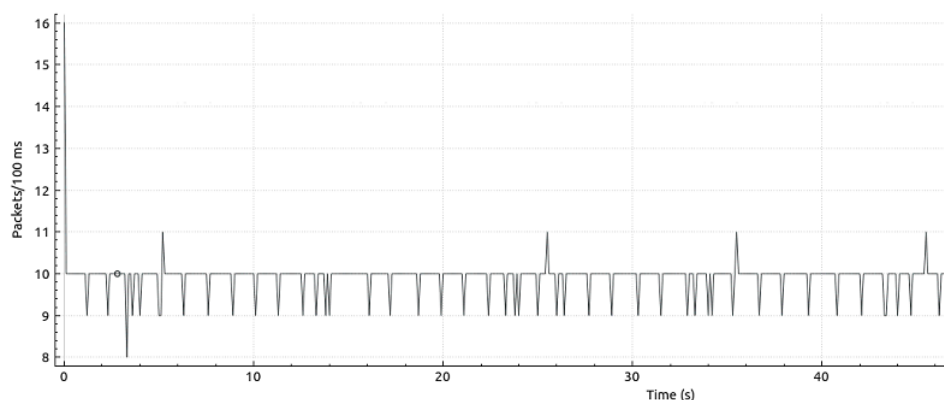
Obr. 6.27: Závislost zpoždění a jeho kolísání na čase pro 2 klienty (video 480p)



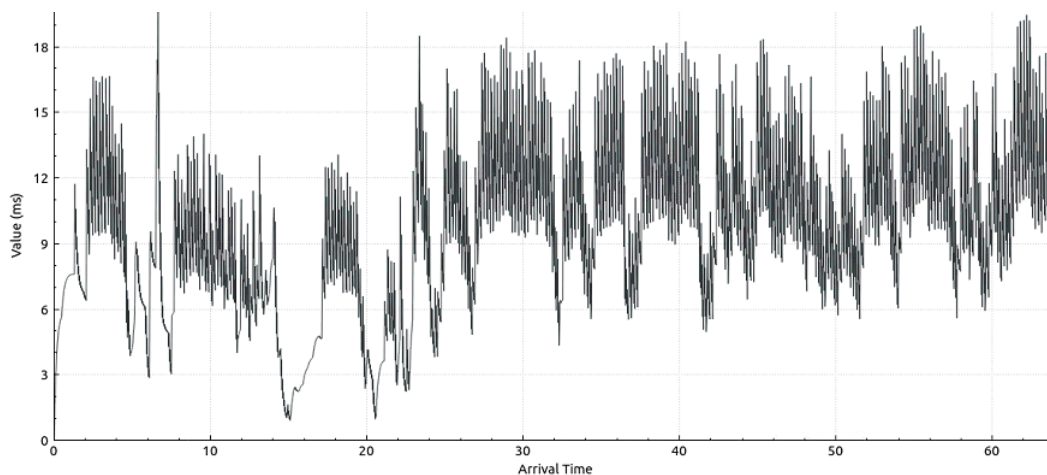
### 6.3.3 RTP stream video 720p

Pro streamování videa 720p byly dosaženy tyto výsledky. Ztrátovost paketů činí 55 %. Na obrázku 6.28 jsou vidět výrazné změny v průběhu počtu paketů. To bylo způsobeno výraznými špičkami v kolísání zpoždění. V této části docházelo k výpadkům přenosu. Během přenosu byly hodnoty zpoždění doručování maximálně 18 ms. Celkový počet přenesených RTP paketů činil 6348 (8696 kB). Pomocí programu Nload byla zobrazena nejvyšší rychlost 1,04 Mbit/s.

Obr. 6.28: Počet přenesených paketů v čase pro video rozlišení 720p



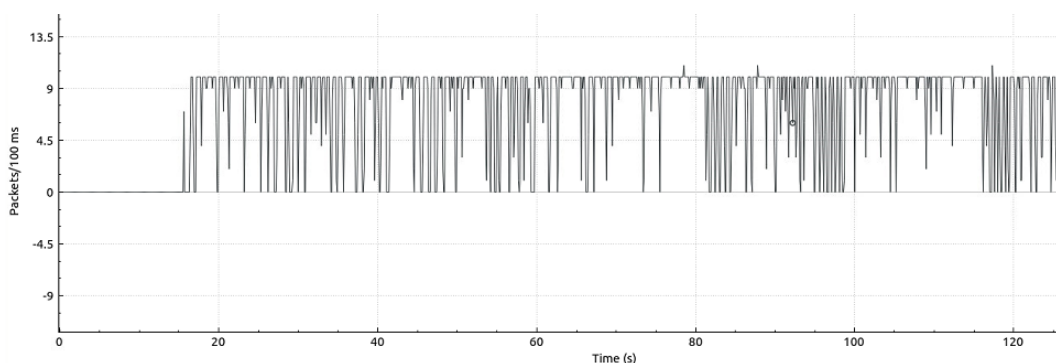
Obr. 6.29: Kolísání zpoždění pro video rozlišení 720p



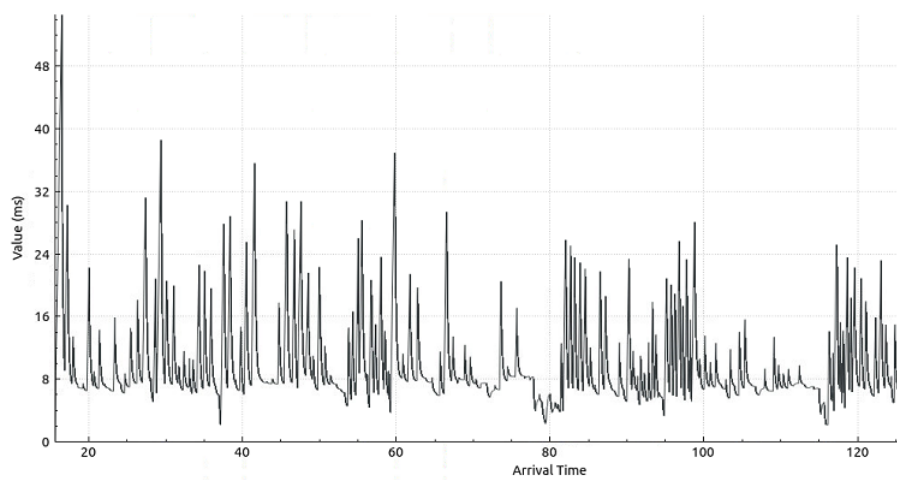
### 6.3.4 RTP stream video 1080p

Pro streamování videa s rozlišením 1080p byly zaznamenány tyto hodnoty. Při přenosu paketů můžeme vidět kolísání odesílání paketů. Celkem bylo přijato 8 613 kB (počet RTP paketů 6287) při průměrné rychlosti 0,213 Mbit/s. Na obrázku 6.30 bylo zaznamenáno výrazné kolísání zpoždění. Velké kolísání zapříčinilo nekvalitní obraz a mělo vliv na ztrátovost paketů, která činí vysokých 80 %. Vysoká ztrátovost způsobila naprosto nečitelný obraz. Tato přenosová technologie tudíž není vhodná pro přenos videa s vysokým rozlišením.

Obr. 6.30: Počet přenesených paketů v čase pro video rozlišení 1080p



Obr. 6.31: Kolísání zpoždění pro video rozlišení 1080p





## 6.4 Porovnání kodeků a jejich přenosových parametrů

Použité nekomprimované video mělo velikost 2,6 GB. Takováto velikost byla příliš velká pro přenos po síti, proto byla nutná komprese dat. Čím vyšší byla účinnost komprimace dat, tím klesal objem dat. Pro detailnější porovnání byly vybrány následující kodeky H.264, MPEG-4 a MPEG-2. Multimediální Full HD přenos byl uskutečněn na gigabitové lince.

### 6.4.1 H.264

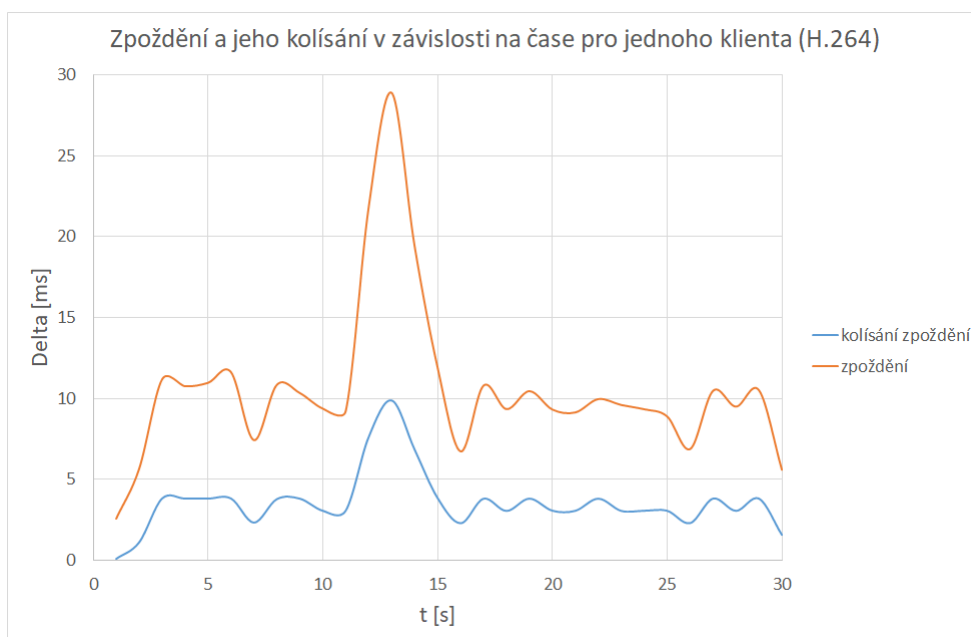
Video mělo tyto vlastnosti. Datová rychlost činila 2721 Kb/s, bitová rychlost byla určena na hodnotě 2,7271 Mb/s. Video bylo dlouhé půl minuty a počet FPS byl 30, celkový počet rámců odpovídal hodnotě 900. Nekomprimované video bylo redukováno z velikosti 2,6 GB na přibližně 10 MB. Komprese trvala téměř 40 sekund.

V prvním případě byla změřena bitová rychlost. Průměrná bitová rychlost se pohybovala okolo hodnoty 2793,4 Kbit/s. Při přenosu byla dosažena nejvyšší přenosová rychlost 5513,7 Kbit/s. Naopak nejnižší přenosová rychlost byla změřena na začátku přenosu, kdy teprve začalo docházet k navazování spojení. Jako druhá veličina bylo změřeno zpoždění a jeho kolísání. Průměrné zpoždění činilo 10,60 ms a průměrná hodnota kolísání zpoždění byla 3,56 ms. Nejvyšší hodnota zpoždění byla zaznamenána na hodnotě 28,92 ms a byla doprovázena vysokým kolísáním zpoždění 9,88 ms. Tato špička byla způsobena snížením bitové rychlosti, tedy nižším datovým tokem videa. Nejnižší hodnota zpoždění a jeho kolísání byla ze začátku přenosu (zpoždění = 2,53 ms a jeho kolísání = 0,06 ms). Celková ztrátovost přenosu byla 1,72 % (138 paketů), tato hodnota se nikterak nepodepsala na kvalitě videa.

Obr. 6.32: Bitová rychlost v čase (kodek H.264)



Obr. 6.33: Zpoždění a jeho kolísání v závislosti na čase (kodek H.264)



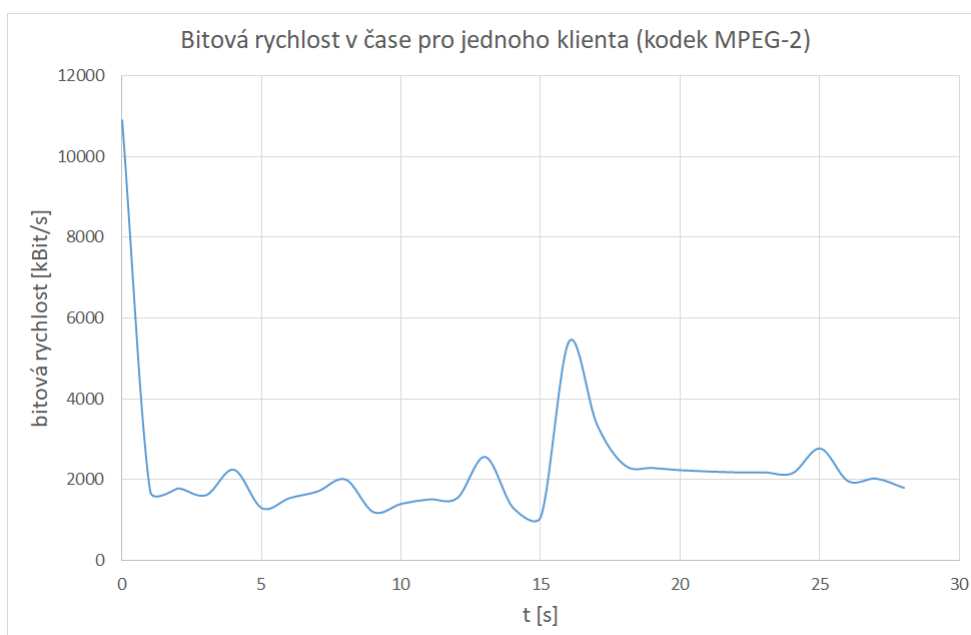
## 6.4.2 MPEG-2

Komprimace pomocí tohoto kodeku trvala 21 sekund. Díky komprimaci byla zredukována velikost na necelých 8 MB (7 878 439 B). Vlastnosti videa jako datová a bitová rychlost byly zaznamenány na hodnotách 2096 Kb/s a 2,1009 Mb/s. Počet

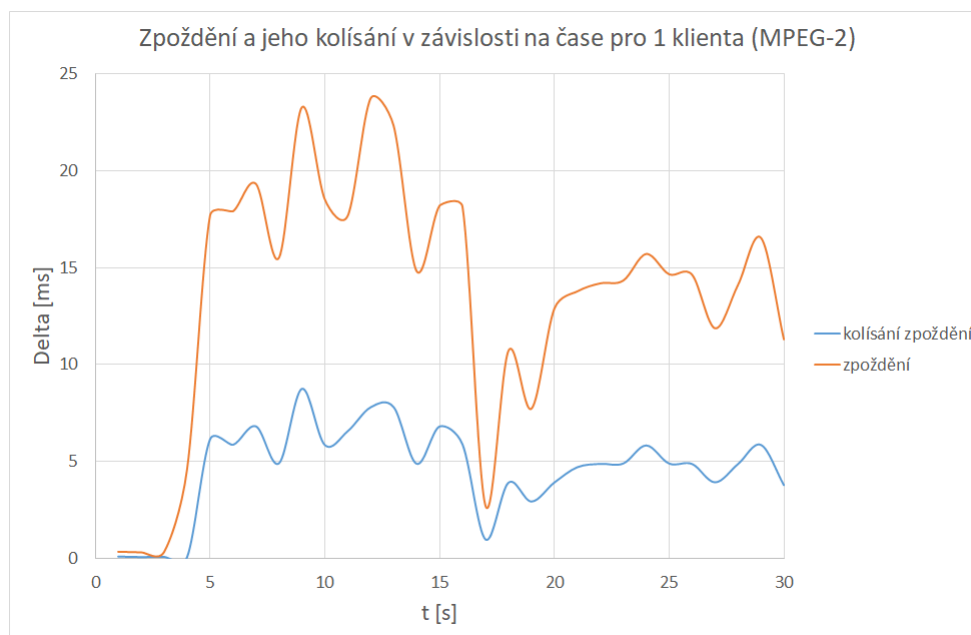
rámců za sekundu byl zaznamenán na hodnotě 30.

Nejvyšší hodnota bitové rychlosti (10905,2 Kbit/s) byla změřena hned na začátku přenosu. Průměrná bitová rychlost činila 2348,85 kBit/s. Nejnižší bitová rychlost byla 1085,1 kBit/s, jelikož se v dané části přenosu nenacházely scény s rychlými pohyby. Stejně jako v případě kodeku H.264 byly zde viditelné špičky v rozmezí 5 až 14 sekund videa jak ve zpoždění, tak v jeho kolísání. Nejvyšší hodnota zpoždění a jeho kolísání činila 23,75 ms resp. 8,72 ms. Poté byl zaznamenán výrazný pokles těchto veličin vlivem změny scény videa a zvýšeným datovým tokem. V první části měření se hodnota zpoždění pohybovala okolo hodnoty 0,30 ms (kolísání zpoždění = 0,02 ms). V tomto měření nebyla zaznamenána žádná ztrátovost paketů.

Obr. 6.34: Bitová rychlost v čase (kodek MPEG-2)



Obr. 6.35: Zpoždění a jeho kolísání v závislosti na čase (kodek MPEG-2)

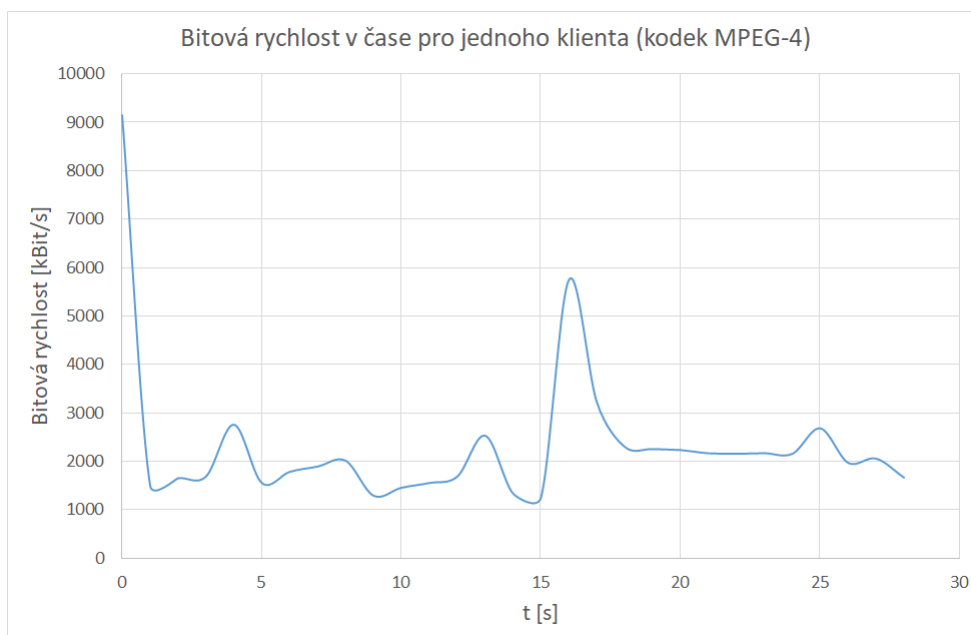


### 6.4.3 MPEG-4

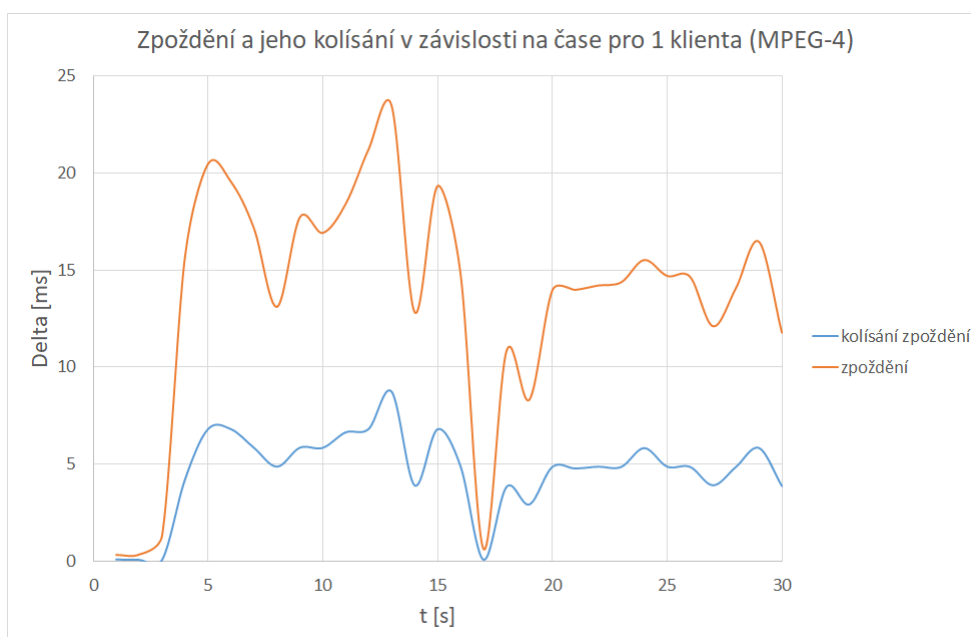
Čas komprimace činil 33 sekund. Objem dat byl snížen na 7,9 MB (7 866 584 B). Celkový počet rámců půlminutového videa byl 900 rámců. Průměrná datová rychlost byla určena na hodnotě 2093 Kb/s a bitová rychlost měla hodnotu 2,09775 Mb/s.

Byla změřena průměrná bitová rychlost a tato hodnota činila 2343,52 kBit/s. Nejvyšší přenosová rychlost byla zaznamenána hned při navázání spojení (9151,6 kBit/s). Nejnižší bitová rychlost byla zaznamenána v půlce měření (15. sekunda), tato hodnota se rovnala 1260,4 kBit/s. Při přepínání scény videa nebylo potřeba vysoké bitové rychlosti. Průměrné zpoždění se rovnalo hodnotě 13,60 ms a průměrné kolísání zpoždění 4,60 ms. Nejvyšší zpoždění činilo 23,44 ms a největší hodnota pro kolísání 8,72 ms. Po této špičce následoval výrazný pokles hodnoty zpoždění a jeho kolísání. Důvodem tohoto poklesu je změna scény videa. V danou chvíli se streamovala náročnější scéna videoobsahu. Nejnižší zpoždění bylo určeno při sestavování spojení (0,34 ms), nejnižší hodnota kolísání se rovnala hodnotě 0,04 ms. Pomocí Wiresharku byla změřena nepatrná ztrátovost 0,16 % (10 paketů).

Obr. 6.36: Bitová rychlost v čase (kodek MPEG-4)



Obr. 6.37: Zpoždění a jeho kolísání v závislosti na čase (kodek MPEG-4)



#### 6.4.4 Souhrnné porovnání

V tabulkách níže jsou zapsány hodnoty pro jednotlivé kodeky. Pro kodek H.264 byla změřena nejvyšší hodnota zpoždění. Průměrná a maximální hodnota zpoždění nabyla identických hodnot jak pro MPEG-2, tak i pro MPEG-4. Největší přenosová

rychlost byla změřena pro kodek MPEG-2, a tato hodnota se rovnala 10905 kBit/s.

Tab. 6.15: Změřené parametry pro kodek H.264

| <b>Změřené parametry pro H.264</b> |                                 |                      |
|------------------------------------|---------------------------------|----------------------|
| <b>Hodnota</b>                     | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná                           | 2793,4                          | 10,60                |
| Max.                               | 5513,7                          | 28,92                |
| Min.                               | 0,696                           | 2,53                 |

Tab. 6.16: Změřené parametry pro kodek MPEG-2

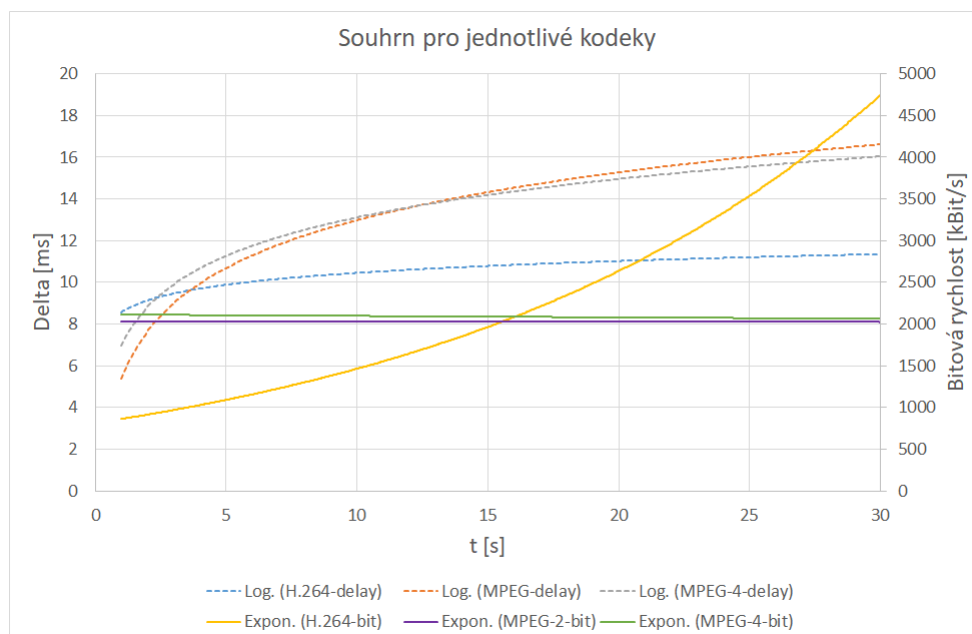
| <b>Změřené parametry pro MPEG-2</b> |                                 |                      |
|-------------------------------------|---------------------------------|----------------------|
| <b>Hodnota</b>                      | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná                            | 2348,85                         | 13,59                |
| Max.                                | 10905,2                         | 23,75                |
| Min.                                | 1085,1                          | 0,30                 |

Tab. 6.17: Změřené parametry pro kodek MPEG-4

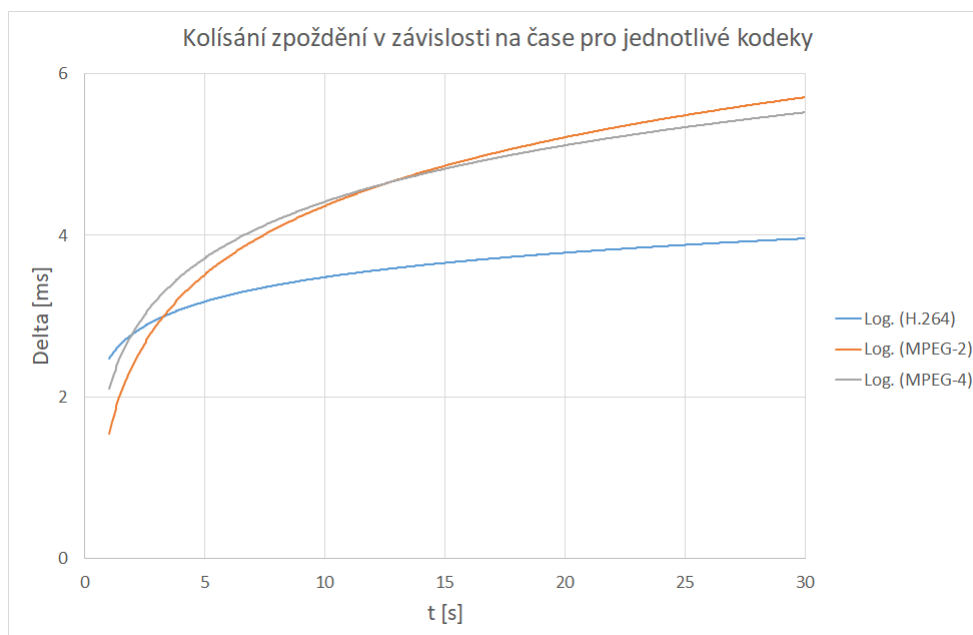
| <b>Změřené parametry pro MPEG-4</b> |                                 |                      |
|-------------------------------------|---------------------------------|----------------------|
| <b>Hodnota</b>                      | <b>Bitová rychlost [kBit/s]</b> | <b>Zpoždění [ms]</b> |
| Průměrná                            | 2343,52                         | 13,60                |
| Max.                                | 9151,6                          | 23,44                |
| Min.                                | 1260,4                          | 0,34                 |

Pro výstižnost a jednoduchost jsou řady grafu vyneseny jako spojnice trendu. Pro bitovou rychlost byla použita exponenciální spojnice trendu a pro zpoždění byla využita logaritmická spojnice trendu. Tato závislost je zobrazena na grafu 6.38. Změřená zpoždění všech kodeků mají tvar rostoucí funkce. Logaritmické spojnice trendu s přibývajícím časem rostou. Logaritmická funkce (H.264) roste od hodnoty 8 ms až do hodnoty 11,3 ms. Výraznější nárůst zpoždění mají logaritmické funkce (MPEG-2 a MPEG-4), a to až skoro do hodnoty 17 ms. Naopak pro bitovou rychlost je zaznamenán největší exponenciální nárůst pro H.264. Tento průběh se pohybuje od hodnoty 1000 kBit/s až k hodnotě 4000 kBit/s. Pro MPEG-2 a MPEG-4 hodnoty rychlosti s přibývajícím časem mírně exponenciálně klesají. Průběhy pro kolísání zpoždění jsou zobrazeny na grafu 6.39. Pro kodeky MPEG-2 a MPEG-4 jsou průběhy téměř identické. Pro tyto kodeky kolísání zpoždění logaritmicky roste (od 2 až po 6 ms) s přibývajícím časem. U kodeku H.264 se kolísání zpoždění logaritmicky zvyšuje mírněji, a to od hodnoty 2 ms až po hodnotu 4 ms.

Obr. 6.38: Souhrnný graf pro bitovou rychlost a zpoždění v čase pro jednotlivé kodeky



Obr. 6.39: Souhrnný graf pro kolísání zpoždění v závislosti na čase pro jednotlivé kodeky



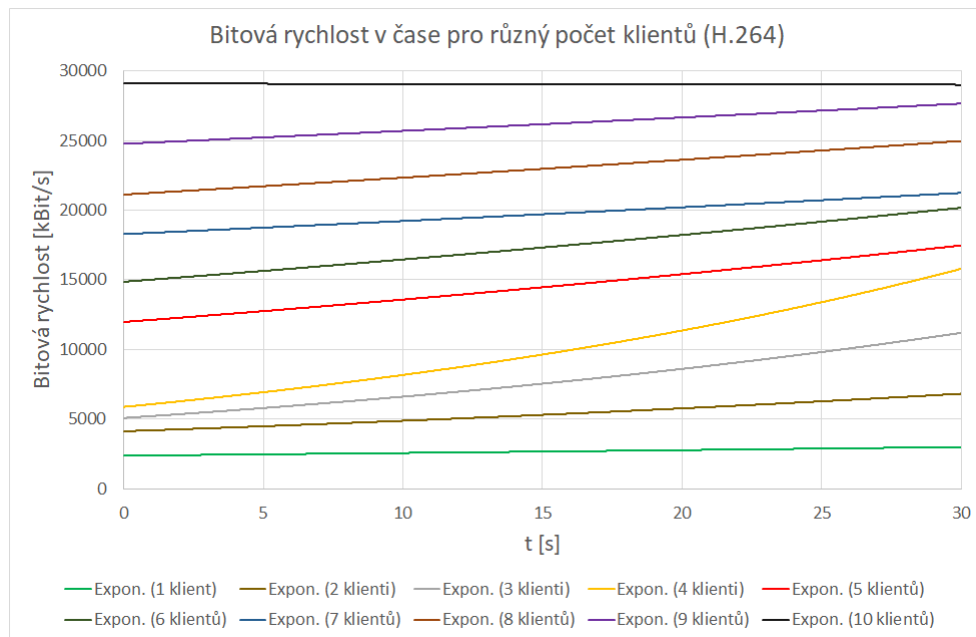
#### 6.4.5 Změření bitové rychlosti pro různý počet klientů

Pro srovnání byla vybrána videa, která byla kódována pomocí kodeků H.264 a MPEG-4. Z grafů je patrné, že se zvyšujícím se počtem klientů byla změřena větší bitová rychlost.

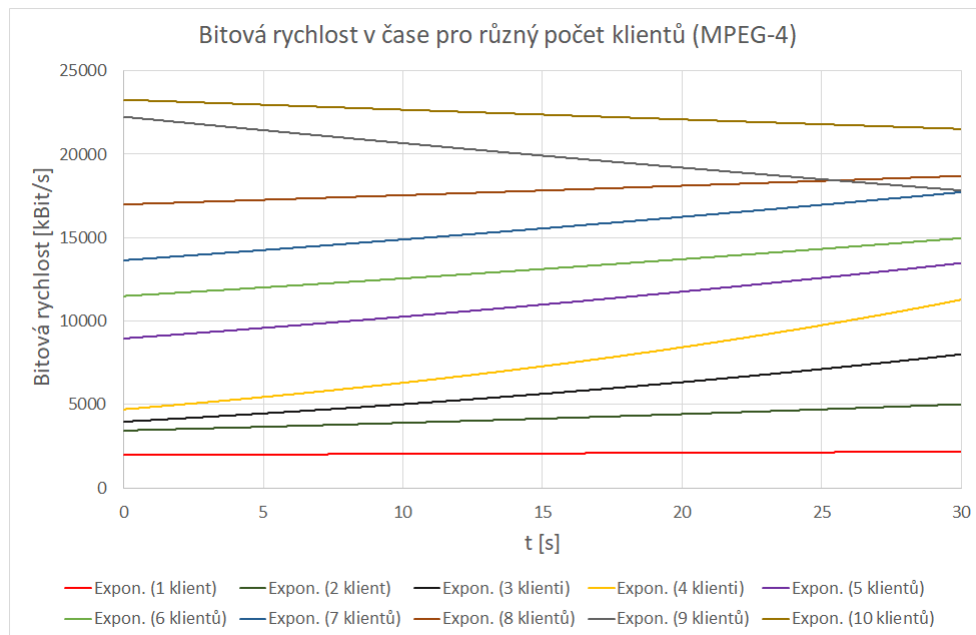
Na grafu 6.40 byla znázorněna bitová rychlost v čase pro kodek H.264. V porovnání s 1 klientem byl pro 10 klientů zaznamenán 90,02 % nárůst bitové rychlosti. Na grafu 6.41 byla znázorněna bitová rychlost v čase pro kodek MPEG-4. Pro tento kodek byly zaznamenány nižší hodnoty přenosových rychlostí. Procentuální nárůst bitové rychlosti při porovnání nejnižšího počtu klientů s nejvyšším činil 90,11 %.



Obr. 6.40: Bitová rychlost v čase pro různý počet klientů (kodek H.264)



Obr. 6.41: Bitová rychlost v čase pro různý počet klientů (MPEG-4)



## 7 Generátor síťového provozu

Pro účel sběru dat byl vytvořen skript, který generoval data dle naměřených hodnot z provozu reálné sítě. Skriptování probíhalo pomocí skriptovacího nástroje Bash skript. Použité hodnoty byly náměrem dvou dnů. Tento náměr obsahoval rozhraní, časové razítko a přesný počet přenesených bitů. Aby data mohla být použita, musel být počet bitů podělen počtem klientů v dané podsíti, tedy dvaceti. Tím byla zátěž rovnoměrně rozložena.

Na prvním řádku je hlavička skriptu, díky ní rozpoznáme, ve kterém jazyce je skript psán. Poté je zde proměnná *I*, která slouží jako index řádku, zatím je používána jako pomocná informace. Následují proměnné, které definují výchozí hodnotu počtu sekund  $\_S=0$ . Dále případný počet sekund je nastaven pro odpočet  $S=\$S$ . Jednou z možností nastavení odpočtu sekund je inicializování proměnné  $\_S$  na nuluovou hodnotu, nebo lze nastavit odpočet přímo v *./packETHcli* pomocí parametru *-t*. Poté byly inicializovány proměnné *SKIP*, *IF\_NAME* a *OUT\_FILEPATH*. Čísla znamenají pozice argumentu. V další části kódu je deklarována podmínka *if*, jež upozorňuje na situaci, ve které by chyběl jeden z parametrů (jméno síťovky nebo cesta k pcap souboru), případně dojde k ukončení skriptu. Uživatel je tak upozorněn, že chybí jeden z klíčových argumentů.

Pomocí cyklu *while read line* se přečte každý řádek v daném souboru. Csv soubor může obsahovat jména sloupců, argumentem *true* je zadáno, aby byl přeskočen první řádek. Součástí cyklu je podmínka, která zajistí vynechání řádku se jmény sloupců. Inkrementuje se index řádku o jeden. Nastaví se hodnota proměnné *SKIP* na *false*, aby se tento krok povedl jen jednou, a pomocí příkazu *continue* se pokračuje od začátku a podmínka je ukončena (*fi*). Odesílají se data z *\$I.* řádku. Všechny čtyři sloupce v řádku jsou procházeny a vytisknuty do proměnné. Čtvrtý sloupec je podělená hodnota odpovídající počtu bitů k odeslání. Znovu se inkrementuje index řádku o jeden. Proběhne zaznamenání hodnot jednotlivých sloupců. Zaznamenaná hodnota čtvrtého sloupce je načtena do příkazového řádku programu *packETH*.

**Parametry generátoru:** *-i* je název síťové karty načten z proměnné *IF\_NAME*, *-m2* mód generování, *-B* počet odesílaných Mbit/s (hodnota předána ze čtvrtého sloupce), *-t* čas odesílání, nebo může být použit argument *-n* počet paketů, poté je načten název pcap souboru z proměnné *OUT\_FILEPATH* a jako poslední je zadáno pořadové číslo paketu v příslušném pcap souboru, jehož hodnota je každým odesláním inkrementována o hodnotu jedna.

V poslední části skriptu se čeká  $\$S$  sekund. Začne odpočet. Pokud  $\$S$  je větší jak 0, proběhne cyklus *while*. Od proměnné  $\$S$  se odečítá hodnota jedna. Po odpočtu se musí počet sekund znovu nastavit na výchozí hodnotu  $S=\_S$ . Při inicializaci hodnoty proměnné na  $\_S=0$  je tento cyklus zbytečný a čas je nastaven přímo

v příkazovém řádku packETHu nebo vůbec. Při definování počtu paketů argumentem -n (v *./packETHcli*) by bylo nastaveno odečítání sekund inicializací proměnné na *\_S=16* tak, aby to odpovídalo časovému razítku v náměru. Poslední řádek kódu znamená, jaký soubor byl vybrán, tedy *sudo ./read\_csv.sh out\_convert.csv true ens3 paket.pcap*.

**PackETH** je nástroj pro generování paketů pro Ethernet, který je dostupný jako GUI nebo pouze jako CLI (příkazový řádek). Je vhodný pro testování výkonu sítě a propustnosti. V grafickém uživatelském prostředí byla vytvořena datová jednotka (pcap soubor), která je odesílána pomocí příkazového řádku *./packETHcli*.

Dále mezi jeho vlastnosti patří:

- vytvoření a odeslání libovolného paketu
- Podporované protokoly: Ethernet II, 802.3, 802.1q
- IPv4, IPv6, ARP
- UDP, TCP, ICMP, IGMP a RTP
- Definice zpoždění, počet paketů k odeslání, odesílání maximální rychlostí
- Načítání pcap souborů, ukládání paketů do pcap souborů

Příkazový řádek umožňuje snadno odesílat pakety ze souboru pcap. Má různé možnosti odesílání.

Existuje 6 různých režimů:

- m1 odesílání jednoho paketu (*./packETHcli -m 1 -i <interface> -f file*)
- m2 odesílání paketu s konstantní rychlostí (*./packETHcli -m 2 -i <interface> -B <Mbit/s> -b <kbit/s> -t <time> -n <počet paketů> -f file*)
- m3 odesílání paketu s proměnnou rychlostí (*./packETHcli -m 3 -i <interface> -t <time> -n <počet paketů> -Z "Startrate Steprate Stoprate" -p <čas mezi odesíláním> -f file*)
- m4 odesílání paketu s proměnnou velikostí (*./packETHcli -m 4 -i <interface> -d/-D <zpoždění mezi pakety> -t <time> -n <počet paketů> -s "startrate steprate stoprate" -f file*)
- m5 odesílání sekvence paketů (*./packETHcli -i <interface> -f <txt soubor> <Mbit/s> -b <kbit/s> -m 5 -t <čas> -S <velikost paketů v B> -a <počet odeslání od do>*)
- m9 režim přijímače (*./packETHcli -i <interface>*), ve kterém lze zachytit a zkontrolovat chyby paketů odeslaných pomocí packETH nebo packETHcli [26]

Výpis 7.1: Bash skript pro čtení csv souboru a generování dat

```
#!/usr/bin/env bash
I=1 _S=0 S=$S SKIP=$2 IF_NAME=$3 OUT_FILEPATH=$4
if [ ! $IF_NAME ] || [ ! $OUT_FILEPATH ]; then
    exit
fi
while read line; do
    if $SKIP ; then
        I=$((I+1))
        SKIP=false
        continue
    fi
    echo --- Odesílání dat z $I. řádku ---
    COL1=$(echo $line | awk -F ";" '{print_1$1}')
    COL2=$(echo $line | awk -F ";" '{print_1$2}')
    COL3=$(echo $line | awk -F ";" '{print_1$3}')
    COL4=$(echo $line | awk -F ";" '{print_1$4}')

    I=$((I+1))
    echo SLOUPEC1: $COL1
    echo SLOUPEC2: $COL2
    echo SLOUPEC3: $COL3
    echo SLOUPEC4: $COL4

    ./packETHcli -i $IF_NAME -m 2 -B $COL4 -t 16
                  -f $OUT_FILEPATH -c $COL1

    echo "-----"
    while [ $S -gt 0 ]; do

        echo -ne "Zbývá_$S_sekund\r"
        sleep 1
        S=$((S-1))
    done

    S=$_S
done < $1
```

Pro hromadné spuštění provozu pro 20 klientů byl vytvořen skript, který umožňuje generování s různými pcap soubory pro jednotlivé koncové stanice. V první části je definována podmínka, aby byly zadány všechny potřebné parametry. Pokud by nebyly zadány, je skript ukončen (*exit*). *For* cyklus prochází složku *PCAPFILES* zadanou z proměnné *PCAP\_FOLDER*. Dochází k posílání těchto souborů a spuštění terminálu s příkazem *-e*, ve kterém se spustí skript pro čtení csv souboru s následným generováním dat. Spouštění tohoto skriptu probíhá pomocí příkazu *sudo ./autopacketh.sh read\_csv.sh out\_convert.csv PCAPFILES true ens3*.

Výpis 7.2: Bash skript pro spuštění generátorů

```
#!/usr/bin/env bash

READ_CSV_SCRIPT=$1
CSV_FILE=$2
PCAP_FOLDER=$3
SKIP=$4
IF_NAME=$5

if [! "$READ_CSV_SCRIPT" ] || [! "$CSV_FILE" ]
|| [! "$PCAP_FOLDER" ] || [! "$SKIP" ] || [! "$IF_NAME" ];
then
    exit
fi

for PCAP_FILE in "$PCAP_FOLDER"/*; do
    echo "Posílání $PCAP_FILE"
    gnome-terminal -e "sh $READ_CSV_SCRIPT $CSV_FILE $SKIP
                        $IF_NAME $PCAP_FILE"
done
```

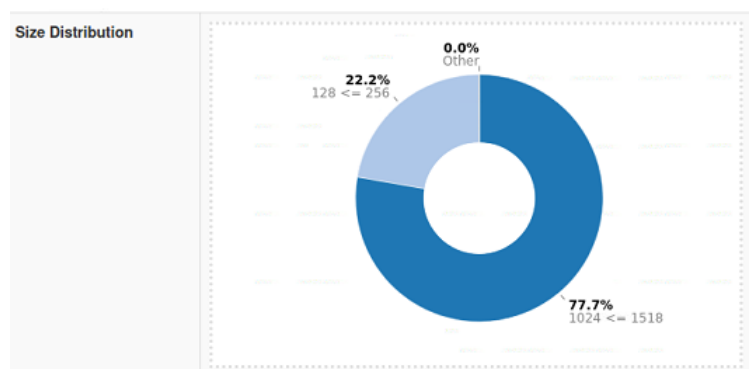
## 7.1 Simulace síťového provozu

Soubor s pakety byl tvořen pomocí grafického uživatelského programu PackETH a také pomocí generátoru Ostinato tak, aby odpovídal provozu na reálné síti. Na každé stanici byly povoleny jednotlivé porty ve firewallu, jelikož Ubuntu stanice mají vstupní komunikaci zakázanou a výstupní povolenou. Z důvodu omezené kapacity disku jednotlivých stanic nebyla data ukládána.

### 7.1.1 UDP simulace

Jako první proběhla simulace UDP provozu a měření maximální propustnosti sítě. Měření probíhalo z pohledu koncové stanice (server). Tato simulace celkově trvala téměř 1 hodinu a 30 minut. Celkem bylo přeneseno 300,34 GB, což odpovídá hodnotě 346 323 319 paketů. Celkově bylo přijato 117,53 GB (240 968 578 paketů). Při takto náročném přenosu docházelo ke ztrátovosti paketů. Celkový počet zahozených paketů byl 942 662. Během simulace byly přenášeny různě velké datagramy. 77,7 % provozu tvořily jednotky velké 1024 až 1518 B. Maximální hodnota byla omezena vzhledem k maximální velikosti MTU. 22,2 % provozu tvoří pakety o velikosti 128 až 256 B.

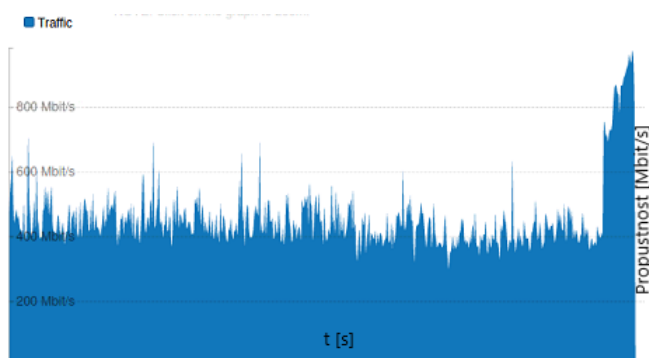
Obr. 7.1: Velikost paketů pro UDP simulaci



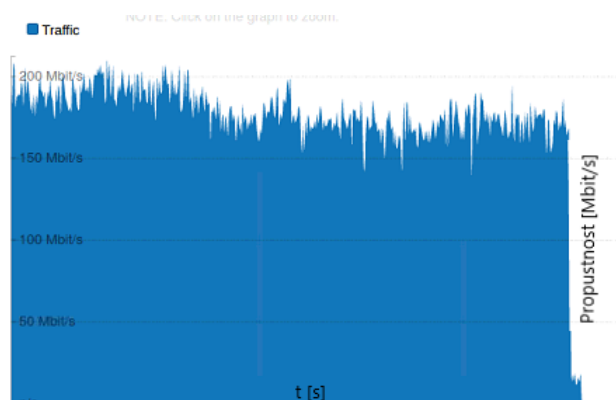
Skript byl spuštěn na stanici server a také na jedné stanici nacházející se v podsíti 172.16.2.0/24. Od stanice server byl provoz generován pro dvacet klientů. Pomocí programu Ntopng byly zobrazeny grafy. Přenosové rychlosti pro upload dosahovaly těchto hodnot (viz graf 7.2). Největší hodnota propustnosti uploadu a v rámci celého měření byla zaznamenána na samotném konci. Tato hodnota činila 954,27 Mbit/s. Nárůst do takto vysoké hodnoty byl způsoben vypnutím skriptu pro download. V jeden okamžik UDP provozu bylo naráz přeneseno 113 202 paketů. Nejvyšší počet zahozených paketů dosahoval hodnoty 2 090.

Pro download byly zaznamenány tyto přenosové parametry. Nejvyšší rychlost downloadu činila 211,87 Mbit/s. Maximální hodnota přenesených paketů dosáhla hodnoty 50 838.

Obr. 7.2: Propustnost linky pro UDP simulaci - upload



Obr. 7.3: Propustnost linky pro UDP simulaci - download

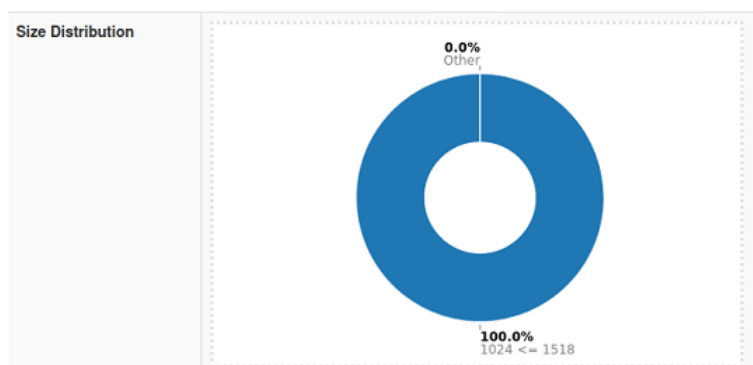


### 7.1.2 RTP simulace

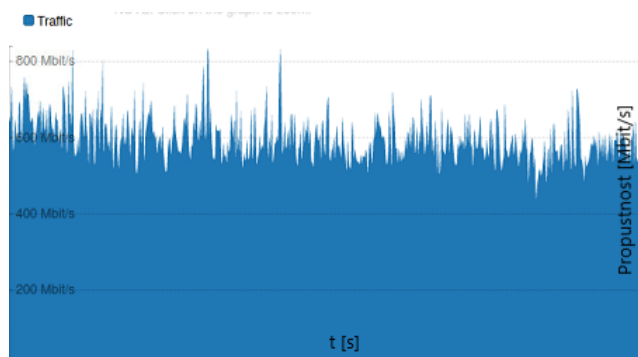
K druhému měření pomocí skriptu byl použit protokol RTP. Tato simulace trvala 1 hodinu a 30 minut. Velikost paketu pro upload i download byla větší jak 1000 B. Tyto velikosti byly zvoleny z toho důvodu, aby přenos odpovídal streamování HD či Full HD videa. Celkový počet přenesených dat činil 375,68 GB. Tento počet odpovídal 327 467 861 paketů. Celkový počet přijatých dat se rovnal hodnotě 167,87 GB (130 014 269 paketů). Ve srovnání s UDP simulací se jedná o značný nárůst přenesených dat. To bylo způsobeno použitím větších přenosových jednotek.

Provoz byl zachytáván na rozhraní ens3 stanice server. Na grafu č. 7.5 jsou viditelné špičky přenosových rychlostí. Tyto špičky dosahovaly hodnot přes 800 Mbit/s. Největší přenosová rychlost byla 832,32 Mbit/s. Téměř pravidelně byly zaznamenány přenosové rychlosti větší jak 500 Mbit/s. Při vyšších přenosových rychlostech byl změřen i vyšší počet přenesených paketů. Při nejvyšší přenosové rychlosti došlo k odeslání 89 647 paketů. Vysoká zátěž sítě a zahlcení routerů způsobily ztrátovost paketů. Celkový počet zahozených paketů se rovnal hodnotě 1 955 722. Maximální množství zahozených paketů v jeden okamžik bylo stanoveno na hodnotě 5 638.

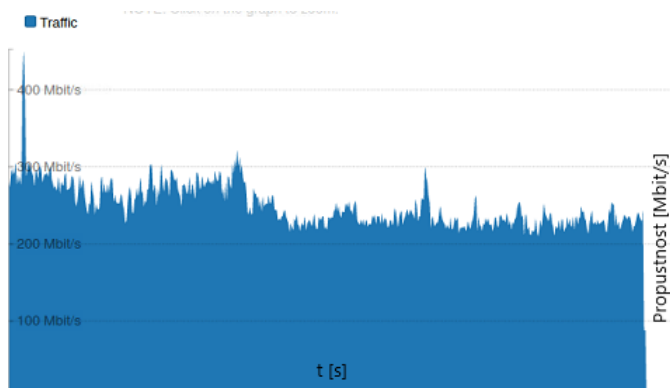
Obr. 7.4: Velikost paketů pro RTP simulaci



Obr. 7.5: Propustnost linky pro RTP simulaci - upload



Obr. 7.6: Propustnost linky pro RTP simulaci - download



Pro download byly zaznamenány tyto hodnoty. Programem Ntopng byla změřena nejvyšší přenosová rychlost a činila 448,69 Mbit/s. Oproti UDP simulaci se jednalo o dvojnásobně větší hodnotu rychlosti, což přímo úměrně odpovídá dvojnásobně větší velikosti paketů (1370 B). Počet přenesených paketů na stanici server přesahoval hodnotu více jak 20 000 paketů za sekundu. Maximální hodnota přenášených paketů byla 28 939.



# Závěr

Tato práce se zabývala měřením kvalitativních parametrů v přístupové síti. V první části byla rozebrána problematika komunikace v počítačové síti. Byly analyzovány jednotlivé vrstvy TCP/IP modelu. Dále byly popsány protokoly, které slouží ke komunikaci, řízení a pro přenos multimediálních dat. Pro zajištění QoS jsou důležité přenosové parametry. Tyto parametry byly blíže specifikovány, resp. jejich vlastnosti a čím jsou ovlivněny. V poslední části teorie byl popsán simulační program GNS3. Tento open source program poskytuje různé možnosti získání nových síťových znalostí a testování sítě s reálnými výsledky. GNS3 nabízí různé možnosti emulace, např. Dynamips, který dokáže emulovat routery od firmy Cisco. Další emulace používaná v praktické části se nazývá QEMU. Programy VirtualBox a VMware nám naopak poskytují virtualizaci zařízení. Pro menší náročnost lze použít k virtualizaci DOCKER. Na závěr teoretické části kapitoly byl jako poslední popsán program Wireshark, který byl používán k zachytávání a vyhodnocování multimediální simulace.

Výstupem bakalářské práce byla charakterizace multimediálních přenosů a vytvoření funkčního síťového generátoru. Nejdříve proběhl návrh sítě, poté konfigurace jednotlivých síťových prvků a koncových stanic. Pro multimediální přenos byla zvolena tři různá přenosová média. Na lince Gigabit Ethernet byla zaznamenána nejnižší ztrátovost. Při přenosu vedením Fast Ethernet docházelo k vyšším ztrátám. Kvalita a funkčnost videa nebyla ovlivňována, pokud se ztrátovost paketů pohybovala v rozmezí 1-5 %. Vyšší ztrátovost měla negativní vliv na výslednou kvalitu videa. Přenos videa v reálném čase vyžaduje dostatečnou šířku pásma. Gigabit Ethernet je nejvhodnější volbou pro tento druh přenosu. Z měření lze usoudit, že jednotlivé přenosové parametry se navzájem ovlivňují. Pokud se v dané části přenosu nacházela scéna s rychlejšími pohyby, mělo to za následek zvýšení bitové rychlosti. Mezi bitovou rychlostí a zpožděním lze pozorovat nepřímou úměrnost. Při zvyšující se bitové rychlosti docházelo ke snižování hodnoty zpoždění. Full HD video dosahovalo nejvyšších hodnot bitové rychlosti a nejnižšího zpoždění. Pro video nejnižší kvality (360p) byly naměřeny nejvyšší hodnoty zpoždění a naopak nejmenší hodnoty bitové rychlosti. Tyto výsledky platily jak pro Gigabit, tak Fast Ethernet. Na lince 10Base-T bylo možné uskutečnit pouze přenos videí 360p a 480p. Pro videa s vyšším rozlišením, jako je HD či Full HD video, byla ztrátovost příliš velká a zamezovala dostatečné kvalitě přenosu. Poslední porovnání proběhlo pro přenos tří videí. Tato videa se odlišovala použitými kodeky při komprimaci. Pro videa s kodeky MPEG-4 a MPEG-2 byly změřeny podobné přenosové parametry. U H.264 byl zaznamenán největší (exponenciální) nárůst bitové rychlosti.

Jako poslední byl popsán vytvořený generátor síťového provozu napsaný v jazyce Bash. Také byl vytvořen skript pro hromadné spuštění generátorů. Sběr dat proběhl

dle předlohy, která obsahovala naměřená data z reálné sítě. Provoz byl měřen po dobu 1,5 hodiny. Následně došlo k vyhodnocení výsledků simulace pro UDP a RTP provoz. Analýza proběhla z pohledu propustnosti, velikosti a ztrátovosti paketů.

# Literatura

- [1] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS*. Druhé aktualizované vydání. Praha: Computer Press, 2000. ISBN 80-7226-323-4.
- [2] PUŽMANOVÁ, Rita. *TCP/IP v kostce*. 3. vydání. České Budějovice: Kopp, 2009. ISBN 978-80-7232-388-3.
- [3] JEŘÁBEK, Jan. *Komunikační technologie*. 2019. Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Technická 12, 616 00 Brno, 2013. ISBN 978-80-214-4713-4.
- [4] ČÍKA, Petr. *Multimediální služby*. Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Purkyňova 118, 612 00 Brno, 2012. ISBN 978-80-214-4443-0.
- [5] SPORTACK, Mark A. *Směrování v sítích IP*. Brno: Computer Press, 2004. ISBN 80-251-0127-4.
- [6] PETERKA, Jiří. Protokol IP. *Earchiv* [online]. Praha: Univerzita Karlova, 2011 [cit. 2020-11-02]. Dostupné z: [http://www.earchiv.cz/1223/gifs/P5\\_27.pdf](http://www.earchiv.cz/1223/gifs/P5_27.pdf)
- [7] SCHULZRINNE, H., R. FREDERICK a V. JACOBSON. RTP: A Transport Protocol for Real-Time Applications. *IETF* [online]. New York, 2003 [cit. 2020-11-15]. Dostupné z: <https://tools.ietf.org/html/rfc3550>
- [8] *ADTRAN: Defining Broadband: Network Latency and Application Performance* [online]. 2009 [cit. 2020-11-18]. Dostupné z: <https://www.adtran.com>
- [9] *What is Network Throughput and How to Measure and Monitor it* [online]. 2020 [cit. 2020-11-15]. Dostupné z: <https://www.ittsystems.com/network-throughput/>
- [10] *Recommendation ITU-T Y.1541* [online]. Geneva, 2011 [cit. 2020-11-02]. Dostupné z: <https://www.itu.int/rec/T-REC-Y.1541-201112-I>
- [11] *ITU-T G.1010* [online]. 2001 [cit. 2020-11-18]. Dostupné z: [itu.int/rec/T-REC-G.1010-200111-I](https://www.itu.int/rec/T-REC-G.1010-200111-I)
- [12] *GNS3 / The software that empowers network professionals*. *GNS3 / The software that empowers network professionals*. [online]. [cit. 2020-11-08]. Dostupné z: <https://gns3.com>

- [13] WELSCH, Chris. *GNS3 Network Simulation Guide*. Birmingham B3 2PB, UK.: Packt Publishing, 2013. ISBN 978-1-78216-080-9.
- [14] NEUMANN, JASON C. *THE BOOK OF GNS3 THE BOOK OF GNS3: BUILD VIRTUAL NETWORK LABS USING CISCO, JUNIPER, AND MORE*. 245 8th Street, San Francisco, CA 94103: No Starch Press, 2015. ISBN 978-1-59327-554-9.
- [15] *QEMU* [online]. [cit. 2020-11-16]. Dostupné z: <https://www.qemu.org/documentation/>
- [16] *VirtualBox* [online]. [cit. 2020-11-16]. Dostupné z: <https://www.virtualbox.org/wiki/Documentation>
- [17] *VMware* [online]. [cit. 2020-11-16]. Dostupné z: <https://www.vmware.com/support/pubs/>
- [18] *Wireshark* [online]. [cit. 2020-11-15]. Dostupné z: <https://www.wireshark.org/>
- [19] SHIMONSKI, Robert. *The Wireshark Field Guide Analyzing and Troubleshooting Network Traffic*. 225 Wyman Street, Waltham, MA 02451, USA: Elsevier, 2013. ISBN 978-0-12-410413-6.
- [20] *Ostinato* [online]. [cit. 2020-11-15]. Dostupné z: <https://ostinato.org/docs/>
- [21] *TecMint Linux The Ideal Blog for Sysadmins and Greeks* [online]. [cit. 2020-11-15]. Dostupné z: <https://www.tecmint.com>
- [22] *IPERF, How to test network Speed,Performance,Bandwidth* [online]. 2013 [cit. 2020-11-15]. Dostupné z: <https://www.slashroot.in>
- [23] *IPERF* [online]. [cit. 2020-11-15]. Dostupné z: <https://iperf.fr>
- [24] *Traceroute: SolarWinds msp* [online]. 2020 [cit. 2020-11-15]. Dostupné z: <https://www.solarwindsmsp.com>
- [25] *Ntop* [online]. Sphinx, 2020 [cit. 2021-02-10]. Dostupné z: <https://www.ntop.org/guides/ntopng/index.html>
- [26] *PackETH* [online]. Miha Jemec, 2018 [cit. 2021-02-10]. Dostupné z: <http://packeth.sourceforge.net>

# Seznam symbolů, veličin a zkratek

**ARP** Adress Resolution Protocol

**BASH** Bourne Again Shell

**BMON** Bandwidth Monitor Network

**CBM** Color Bandwidth Meter

**Cisco IOS** Cisco Internetwork Operating System

**CLI** Command Line Interface

**CPU** Central Processing Unit

**DHCP** Dynamic Host Configuration Protocol

**ES** Elementary Stream

**FPS** Frames Per Second

**GNS3** Graphical Network Simulator 3

**GUI** Graphical user interface

**HTTP** Hypertext Transfer Protocol

**ICMP** Internet Control Message Protocol

**IGMP** Internet Group Management Protocol

**IP** Internet Protocol

**IPv4** Internet Protocol version 4

**IPv6** Internet Protocol version 6

**ISO/IEC** International Organization for Standardization/International  
Electrotechnical Commission

**ITU-T** International Telecommunication Union

**KVM** Kernel-based Virtual Machine

**LAN** Local Area Network

**MAC** Media Access Control

**MPEG** Moving Picture Experts Group

**MTU** Maximum Transmission Unit

**NAT** Network Address Translation

**OS** Operating system

**OSPF** Open Shortest Path First

**OTT** Over-The-Top

**PAT** Program Association Table

**PES** Packetized Elementary Stream

**PID** Program Identifier

**Ping** Packet internet groper

**PMT** Program Map Table

**QoE** Quality of Experience

**QoS** Quality of Service

**RAM** Random-access memory

**RTP** Real-Time Protocol

**RTT** Round Trip Time

**TCP** Transmission Control Protocol

**TCP/IP** Transmission Control Protocol/Internet Protocol

**TTL** Time To Live

**UDP** User Datagram Protocol

**VLC** VideoLAN Client

**VM** Virtual machine

**VPCS** Virtual PC Simulator

**WAN** Wide Area Network

**VT-x** Intel virtualization